



Universidad
Carlos III de Madrid

**UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**

**INGENIERÍA TÉCNICA DE TELECOMUNICACIONES
TELEMÁTICA**

PROYECTO FIN DE CARRERA

**SISTEMA DE GESTIÓN DE
ACTIVIDADES EN AUDITORIOS**

AUTOR: Raquel Fernández Fraile

DIRECTOR: Jesús Miguel Gago Mejías

LEGANÉS, JUNIO 2010



ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	5
1.1 – Motivación.....	5
1.1.1 – Sistemas de publicación Auditorio	6
1.1.2 – Identificación de necesidades	6
1.2 – Objetivos del proyecto.....	7
1.3 – Estructura del documento	8
 CAPÍTULO 2. ESTADO DEL ARTE DE LA TECNOLOGÍA.....	 9
2.1 – Herramientas informáticas de gestión de espectáculos.....	9
2.2 – Estudio de diferentes alternativas aplicables.....	9
 CAPÍTULO 3. ANÁLISIS DEL SISTEMA	 11
3.1 – Definición del sistema	12
3.1.1 – Determinación del alcance del sistema	12
3.1.2 – Elección de la metodología, estándares y normas de desarrollo	14
3.2 – Especificación de requisitos	15
3.3 – Análisis de Casos de Uso	25
3.4 – Análisis de Clases.....	30
 CAPÍTULO 4. DISEÑO DEL SISTEMA	 34
4.1 – Definición de la arquitectura del sistema	35
4.1.1 – Arquitectura física	35
4.1.2 – Arquitectura lógica	37
4.1.3 – Relación entre arquitectura física y lógica	39
4.1.4 – Elección del entorno tecnológico	41
4.1.5 – Identificación de los subsistemas	43
4.2 – Diseño de clases detallado	44
4.3 – Diseño físico de datos.....	50
 CAPÍTULO 5. PLANIFICACIÓN Y PRESUPUESTO.....	 52
5.1 – Planificación del proyecto	53
5.1.1 – Actividades.....	53
5.1.2 – Resumen del proyecto	60
5.1.2 – Diagrama de Gantt.....	63

5.2 – Valoración económica del proyecto.....	64
CAPÍTULO 6. CONCLUSIONES Y FUTURAS LÍNEAS DE DESARROLLO	65
6.1 – Conclusiones	66
6.2 – Futuras líneas de desarrollo.....	68
CAPÍTULO 7. BIBLIOGRAFÍA	70
ANEXO A. TECNOLOGÍAS EMPLEADAS	70
A.1 – Plataforma Java.....	70
A.1.1 – JAVA 2 Estándar Edition.....	72
A.1.2 – JAVA 2 Enterprise Edition.....	72
A.2 – Arquitectura J2EE.....	73
A.3 – Apache Jakarta Tomcat.....	75
A.3.3 – Scripts de Tomcat.....	76
A.3.4 – Ficheros de configuración de Tomcat	77
A.4 – Oracle.....	77
A.4.1 – Introducción	77
A.4.2 – Estructura de la base de datos	78
A.4.3 – Estructura de memoria y procesos.....	78
A.4 – UML (Lenguaje de modelado unificado)	82
ANEXO B. MANUAL DE USUARIO	87
B.1 –Acceso a la aplicación	87
B.2 – Gestión de espectáculos	88
B.3 – Gestión de contenidos	93
B.4 – Gestión de eventos.....	94
B.5 – Gestión de páginas.....	96
B.6 – Gestión de usuarios.....	100
B.6 – Cerrar sesión.....	103
B.8 – Búsquedas	103

CAPÍTULO 1

INTRODUCCIÓN

El proyecto “Sistema de gestión de actividades en auditorios” consiste en la creación de un sistema de gestión para la publicación en web de los distintos espectáculos y actividades de auditorios o teatros.

1.1 Motivación

La elección de este proyecto, parte de la determinación de demostrar que un estudiante puede ofrecer a la Universidad, una buena alternativa para el sistema de gestión de información del Auditorio de la UC3M de una forma fácil y dinámica, sin que sea necesaria la contratación de un servicio externo.

Para ello, esta aplicación deberá ser sencilla, práctica y completa para poder mostrar la información actualizada sin necesidad de disponer de un programador. Además ofrecerá la ventaja de que las páginas puedan ser modificadas por cualquier persona desde cualquier lugar y en cualquier momento.

El objetivo de este proyecto es por tanto, facilitar a la Universidad una herramienta gratuita y personalizada que responda a las necesidades de difusión electrónica del Auditorio. Su desarrollo se ha realizado teniendo en cuenta las características reales de este espacio escénico, sus necesidades de información (distintos espacios, distintos precios, actividades de distinta naturaleza, etc.), y de los propios espectáculos. De esta manera, crear una aplicación lo más útil y eficaz posible, ajustándose al máximo a la demanda real.

Este sistema podría aplicarse a otros espacios, auditorios y teatros que programen espectáculos de forma continua y tengan necesidad de difundirlos sin necesidad de grandes ajustes.

1.1.1. Sistemas de publicación de Auditorio

Desde los orígenes del Auditorio de la UC3M la información de sus espectáculos se mostraba a través de páginas HTML y css (hojas de estilo en cascada). Esto hacía que cada página mostrada en su web tuviera que ser creada una a una repitiendo en muchas de ellas información común, lo que hacía que hubiera una gran carga de trabajo repetitivo difícil de mantener, ya que el tiempo que se le tenía que dedicar era excesivo y los resultados no eran en muchos casos ni inmediatos ni óptimos.

Este trabajo era realizado por un programador, pero el servicio no podía seguir contando con esta figura y se hizo necesaria la búsqueda de una alternativa. En la actualidad se cuenta con una página realizada y mantenida por una empresa externa.

1.1.2. Identificación de necesidades

Este proyecto propone una aplicación basada en Servlets y jsp utilizando una base de datos de Oracle que permite introducir la información general y de espectáculos a través de sencillos formularios.

También mejora la información de incidencias, dándole inmediatez a la publicación de avisos importantes y urgentes, como por ejemplo una ampliación del horario de taquilla o aforo completo, de manera que se pudieran controlar las fechas de publicación y baja y conseguir así que la información esté siempre actualizada.

Por último también se permitirá hacer búsquedas de espectáculos que ya se han realizado o que están programados para el futuro, aunque no aparezcan publicados en la programación actual de la temporada. Esta información puede resultar muy útil para los gestores del Auditorio por dos razones:

- ➡ Se pueden hacer consultas internas de contenido de espectáculos que ya se han representado y rescatar información útil y práctica
- ➡ Se permite al público ver el tipo de programación del Auditorio y hacer todo tipo de búsquedas.

1.2. Objetivos del proyecto

Los objetivos que debe cumplir esta aplicación son:

- **Gestión de información de espectáculos**, permitiendo crear páginas de espectáculos nuevos con toda su información eligiendo la temporada a la que pertenecen, calificación por edades, género y compañía que representa el espectáculo. Además se debe poder añadir todos los precios diferenciados por zonas y perfiles del público, las fechas de representación del espectáculo, y las imágenes y descargas propias de éste para mostrarlo al público de la forma más atractiva posible.
También debe permitir la eliminación de espectáculos que dejen de estar en la programación y la modificación de los datos introducidos por cambios posteriores en la programación.
- Permitir hacer **búsquedas de espectáculos** filtrándolos por categorías, géneros, calificaciones por edades, fechas y compañías. Permitiendo también búsquedas de espectáculos anteriores.
- Gestión de información de **eventos y avisos** relacionados con el Auditorio y mostrarlos en su página principal.
- **Gestión de categorías, géneros, calificaciones por edades y compañías**: se podrá agregar, borrar o modificar cualquiera de estos campos comunes a todos los espectáculos.
- **Gestión de páginas** de información general, permitiendo la creación de páginas o subpáginas nuevas y la modificación o eliminación de las ya creadas. Aparte de la información, también se debe poder gestionar dentro de cada página el orden de los enlaces y las subpáginas.
- **Publicación de la programación actualizada del Auditorio** mostrando los próximos espectáculos previstos.
- **Publicación de otras informaciones relativas al Auditorio**

1.3. Estructura del documento

Capítulo 1. Introducción. Es el presente capítulo, donde se explica la motivación por la cual surge este proyecto y, en rasgos generales, los objetivos que se pretenden conseguir.

Capítulo 2. Estado del Arte de la Tecnología. Es el capítulo que recoge una descripción de algunas aplicaciones de tratamiento de espectáculos existentes en el mercado.

Capítulo 3. Análisis del sistema. Con este capítulo se pretende dar una definición a alto nivel del sistema, identificando el entorno, y especificando la funcionalidad mínima que debe presentar al usuario.

Capítulo 4. Diseño del sistema. En él se incluye el desarrollo, en el dominio de la solución, del modelo conceptual elaborado en el capítulo anterior, a más bajo nivel, especificando la arquitectura que se utilizará en su desarrollo y la definición detallada de los subsistemas que lo componen.

Capítulo 5. Planificación del proyecto y recursos empleados. Incluye la planificación del proyecto, y el coste total que el proyecto ha supuesto.

Capítulo 6. Conclusiones. En este capítulo se revisan los objetivos iniciales respecto a los resultados obtenidos tras la elaboración de este trabajo, así como posibles mejoras y futuras líneas de trabajo.

Capítulo 7. Bibliografía. Capítulo que enumera la documentación utilizada y citada en este trabajo.

ANEXO A. Tecnologías empleadas. En este anexo se incluye una introducción a las distintas tecnologías empleadas en el desarrollo del sistema.

ANEXO B. Manual de usuario. Incluye una descripción detallada del interfaz del sistema e incorpora la especificación de uso de las distintas ventanas de la aplicación para facilitar su manejo.

CAPÍTULO 2

ESTADO DEL ARTE DE LA TECNOLOGÍA

2.1. Herramientas informáticas de gestión de espectáculos

Cuando una organización determina que debe introducir un nuevo componente en sus sistemas de información, tiene varias posibilidades: o bien decidirse por un desarrollo a medida, que cubra exactamente las necesidades de la organización, o bien por la adquisición de un paquete comercial ya desarrollado.

Cada espacio escénico presenta una arquitectura, programación, organización y características estructurales diferentes, lo que hace difícil que existan productos estándar en este campo.

Por este motivo, actualmente no existen en el mercado paquetes comerciales que permitan gestionar y/o publicitar espectáculos. Por lo que cada cliente contrata aplicaciones hechas a medida en función de sus necesidades.

2.2. Estudio de diferentes alternativas aplicables

El desarrollo de cualquier producto debe estar precedido por un análisis de necesidades, costes y viabilidad, que permita garantizar que el desarrollo del proyecto es factible y que los costes pueden ser asumidos por la organización.

La gran ventaja de los productos de desarrollo interno es que se adaptan mejor a las necesidades de la organización frente a los paquetes comerciales estándar. Además, disponer del código fuente de la aplicación, permite corregir errores y hacer mejoras sin depender de un tercero ajeno a la organización. Sin embargo, esta solución suele implicar una dilatación en los plazos de entrega.

Buscando entre varios espacios escénicos se ha visto cómo las diferentes páginas visitadas han sido desarrolladas de forma personalizada por empresas especializadas en el diseño de este tipo de aplicaciones. En la tabla 2.2 podemos ver varios ejemplos.

Espacio Escénico	Estudio de diseño
Auditorio Universidad Carlos III de Madrid	Pentaprixma
Teatro de Madrid	Gorricho/estudio de diseño
Teatro Romea	The Useful Company
Teatro Cervantes	Takeone dsgn

Tabla 2.2. Relación de espacios escénicos con la empresa desarrolladora de su aplicación

Tras lo expuesto anteriormente, vemos que la utilización de este Proyecto Fin de Carrera es una buena alternativa al sistema de gestión actual para el Auditorio de la Universidad Carlos III de Madrid, ya que además de tener todos los beneficios de un producto de desarrollo interno implica un coste menor para la Universidad.

CAPÍTULO 3

ANÁLISIS DEL SISTEMA

En este capítulo se realiza el proceso de Análisis del Sistema a desarrollar de acuerdo con los requisitos previstos para obtener una especificación detallada que satisfaga la información de los usuarios y sirva como base posterior para el diseño del sistema.

Para realizar este análisis se efectúa una definición general del sistema, delimitando su alcance, estableciendo los interfaces con otros sistemas e identificando a los usuarios representativos del mismo. De esta forma, se indica qué procesos pertenecen al ámbito del sistema y se identifican las entidades externas que le aportan o reciben información. Este análisis queda establecido a través de los Diagramas de Casos de Uso detallados en este capítulo.

Asimismo, se obtiene un modelo conceptual identificando el conjunto de clases y las relaciones entre ellas, con el fin de establecer los objetos del dominio sobre los cuales se llevarán a cabo los procesos especificados en los Diagramas de Casos de Uso.

3.1. DEFINICIÓN DEL SISTEMA

Antes de realizar en detalle la especificación de requisitos software, daremos una visión general del sistema delimitando su alcance e identificando a los usuarios representativos.

3.1.1. Determinación del Alcance del Sistema

En esta tarea se detalla de forma general el contexto del sistema, especificando los procesos que pertenecen a su ámbito, e identificando las entidades externas que interactúan con él, con el fin de delimitar el sistema de información.

Como se ha mencionado en capítulos anteriores, el sistema a desarrollar pretende llevar a cabo la publicación de espectáculos, eventos e información por parte de cualquier teatro o auditorio. Su finalidad por tanto es lograr almacenar la información de forma estructurada, para que tanto los administradores de la información como los usuarios simples puedan acceder a ella de una manera sencilla.

A la aplicación se accede de forma independiente, mediante la autenticación de usuarios contra el Directorio Corporativo de la Universidad donde se encuentran los datos de los usuarios del sistema. El resto de la información se almacena en una base de datos propia, que actúa como repositorio de la información de la aplicación. En esta base de datos se registrarán los espectáculos, la información relativa a la sala, los temas de clasificación, los usuarios y sus perfiles de acceso a la aplicación, los eventos registrados como información para el público, etc. Los diferentes perfiles de usuario que pueden acceder al sistema son los siguientes:

Usuario Simple

Usuario al que sólo se permite la consulta de espectáculos, información general e incidencias. Para ello tendrá acceso restringido a la información permitiendo

únicamente la consulta de aquella información a la que esté autorizado a acceder. Para este usuario no será necesaria la validación ya que dicha información es pública.

Para este perfil de usuario no hace falta ningún conocimiento técnico ni del sistema ya que su uso será sólo para consulta de la información que sea de su interés.

Usuario Avanzado

La tarea del usuario avanzado es la introducción de información de espectáculos en la Base de Datos para poder mostrarla posteriormente a todos los usuarios. Además tendrá permisos para gestionar las incidencias producidas así como toda la información general del Auditorio. Además, también será gestor de las categorías de búsqueda.

Este usuario deberá tener un conocimiento completo del sistema, por lo que deberá leer el manual de usuario para conocer toda la funcionalidad de la aplicación.

Administrador

Usuario encargado de gestionar los permisos y perfiles de cada usuario dentro de la aplicación. El administrador tendrá acceso total a la información almacenada en la Base de Datos.

3.1.2. Elección de la metodología, estándares y normas de desarrollo

La metodología utilizada para realizar el desarrollo de este proyecto es Métrica versión 3. Si bien, en virtud de las características específicas de la aplicación, no se han llevado a cabo en su totalidad todas las actividades especificadas en la metodología para el desarrollo de una aplicación orientada a objetos, se han incluido en el proyecto aquellas más significativas por aportar mayor información de interés a las fases de análisis y diseño del sistema.

Para el modelado del sistema se utiliza el Lenguaje de Modelado Unificado (*UML*), basado en el paradigma de la orientación a objetos. El objetivo central de este lenguaje es abstraer cualquier tipo de sistema mediante su representación gráfica a través de diagramas según varias perspectivas o vistas que abarcan la totalidad del proyecto. Los diagramas generados a partir de los procesos son artefactos que están incluidos en las distintas fases de desarrollo del sistema.

En cuanto al método de desarrollo, cabe destacar que el sistema se basa en el ciclo de vida software conocido como *aproximación en espiral*, siguiendo un enfoque *top-down*. Este ciclo de vida exige un enfoque sistemático, ya que en cada ciclo de la espiral se realiza una parte del desarrollo total, siguiendo la secuencia de *Planificación*, donde se establece el contexto de desarrollo, decidiendo qué parte del mismo se abordará en este ciclo; *Análisis de Riesgo*, donde se evalúan las diferentes alternativas, eligiendo la mejor; *Ingeniería*, donde se realiza el análisis, diseño, codificación y pruebas del sistema; y *Evaluación*, donde se analizan los resultados de la fase de ingeniería y cuyo resultado se utiliza como información de entrada para la planificación del ciclo siguiente [Pre93].

3.2. ESPECIFICACIÓN DE REQUISITOS

En este apartado se lleva a cabo la definición, análisis y validación de los requisitos a partir de las necesidades de los usuarios, con el fin de obtener un catálogo detallado de los mismos.

Dividiremos esta especificación en tres partes:

- **Requisitos de datos**, también denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Éstos requisitos responden a la pregunta de qué información debe almacenar y administrar el sistema.

R001	Espectáculos
Descripción	Conjunto de información con entidad propia que incluye todos los datos que se quieren mostrar al usuario sobre el espectáculo.
Datos	Identificador
	Título
	Subtítulo
	Descripción breve
	Información
	Elenco Artístico
	Equipo Técnico
	Duración
	Precio/s
	Descripción venta de entradas
	Categoría
	Compañía
	Género
	Curso
	Calificación por edades

Imágenes
Descargas
Fecha/s de representación

R002	Eventos
Descripción	Información relevante y/o urgente de cualquier temática destacable para el público. Se mostrará en la página de inicio de la aplicación.
Datos	Identificador
	Nombre
	Enlace
	Imagen
	Fecha de publicación
	Fecha de baja
	Sectores a ocupar dentro de la página

R003	Páginas de contenido general
Descripción	Muestra información general del teatro y/o auditorio acerca de otros servicios ofrecidos al público además de los propios espectáculos.
Datos	Identificador
	Título
	Contenido
	Imagen
	Nivel en el que se sitúa la página dentro del árbol de páginas
	Posición de la página respecto al resto con la misma página padre
	Identificador de la página padre
	Indicador de si la página contiene subpáginas

R004 Noticias	
Descripción	Información dedicada a la prensa para publicitar un espectáculo. Podrá ser de dos tipos, bien en forma de nota de prensa o como noticia desarrollada.
Datos	Identificador
	Titular
	Resumen
	Información
	Fecha de publicación de la noticia
	Fecha de baja de la noticia si se quiere dejar de mostrar
	Mostrar en página home
	Identificador de descarga

R005 Perfiles	
Descripción	Almacenará los permisos de los diferentes tipos de usuarios que podrán acceder al sistema.
Datos	Identificador
	Nombre de usuario
	Contraseña
	Rol
	Usuario en Servicio de Directorio

R006 Descargas	
Descripción	Almacena la información necesaria para que el usuario pueda descargarse cualquier archivo público existente en el sistema.
Datos	Identificador
	Título
	url
	Tamaño
	Identificador de tipo de descarga
	Identificador de formato de descarga

- **Requisitos de interfaz** también llamados requisitos de interacción o de usuario. Responden a la pregunta de cómo va a interactuar el usuario con la aplicación.

R007 Perfiles de usuario	
Descripción	<p>La aplicación debe gestionar de forma diferenciada los siguientes perfiles de usuarios:</p> <ul style="list-style-type: none">- Simple.- Avanzado.- Administrador. <p>Cada perfil tendrá un determinado nivel de acceso a las diferentes funcionalidades de la aplicación. Para cada funcionalidad se especificará que perfiles pueden acceder.</p>

R008 Inicio de sesión	
Descripción	Los usuarios avanzado y administrador, deberán acceder a la aplicación a través de un sistema de autenticación mediante nombre de usuario y contraseña. Éstas se deberán guardar cifradas en la aplicación.

R009	Cierre de sesión
Descripción	Las sesiones iniciadas deben cerrarse bien por parte del usuario o automáticamente tras un periodo de inactividad.

R010	Aplicación multiusuario
Descripción	La aplicación debe permitir el acceso múltiple de usuarios de cualquier perfil.

R011	Cliente Web
Descripción	La aplicación podrá usarse al menos con los navegadores web más extendidos: Microsoft Internet Explorer y Mozilla Firefox. No se requerirá la instalación de ningún software adicional por parte de los usuarios.

R012	Accesibilidad
Descripción	<p>Toda la aplicación deberá ser accesible cumpliendo las recomendaciones de accesibilidad WAI (Web Accessibility Initiative) del consorcio Web (W3C).</p> <p>Los niveles de cumplimiento propuestos por la Iniciativa para la Accesibilidad en la Web (WAI) varían desde el 1 en todas las páginas hasta el 3 en muchas de ellas.</p>

- ➡ **Requisitos funcionales** especifican las funciones que ha de desempeñar la aplicación.

R013	Identificadores
Descripción	Cada espectáculo, página o incidencia estarán identificados unívocamente por un identificador numérico.

R014	Títulos
Descripción	Cada espectáculo, página o incidencia tendrá asociado un título que será representativo del tema y contenido que trate.

R015	Gestor de espectáculos
Descripción	La aplicación permitirá insertar nuevos espectáculos, y modificar, eliminar y consultar los ya creados.

R016	Gestión de contenidos
Descripción	Los usuarios avanzado y administrador podrán modificar, agregar y eliminar el contenido de los campos de búsqueda de espectáculos (categoría, calificación, género, perfiles de público, compañía...)

R017	Campos de búsqueda
Descripción	Los espectáculos tendrán como obligatorios los campos utilizados en la búsqueda de espectáculos para que todos ellos estén incluidos en cualquier consulta hecha a la aplicación.

R018	Ficheros
Descripción	Todos los ficheros de descarga asociados a los espectáculos estarán localizados en el servidor.

R019	Imágenes
Descripción	Cada espectáculo deberá tener asociadas dos imágenes, una para los listados de espectáculos de la página de inicio de la de programación y otra para la ficha del espectáculo.

R020	Inserción de categorías, géneros, calificaciones y compañías
Descripción	Al insertar o modificar un espectáculo se podrá añadir categorías, géneros, calificaciones y compañías que aún no estuvieran añadidos en la base de datos pudiéndose reutilizar para otros espectáculos quedándose así agrupados a la hora de realizar una búsqueda.
R021	Fechas
Descripción	A cada espectáculo se le podrán asignar una o varias fechas y horas, lo que permitirá además de mostrar todas las representaciones de un espectáculo al público, ordenar la programación cronológicamente.
R022	Enlaces de interés
Descripción	Cada espectáculo podrá tener asociado uno o varios enlaces de interés, los cuales se podrán guardar en la aplicación para utilizarlos en otros espectáculos.
R023	Precios
Descripción	Cada espectáculo puede tener asociados varios precios. Los precios se pueden establecer por zonas del teatro y por tarifas. El número de zonas del espacio se podrán definir en cualquier momento.
R024	Inserción de tarifas de públicos nuevos
Descripción	Si al añadir o modificar los precios de un espectáculo necesitamos una tarifa que aún no se encuentra en la base de datos podremos añadirlo quedando guardado para poder reutilizarlo en los siguientes espectáculos.

R025 Inserción de categorías, géneros, calificaciones y compañías	
Descripción	Al insertar o modificar un espectáculo se podrá añadir categorías, géneros, calificaciones y compañías que aún no estuvieran añadidos en la base de datos pudiéndose reutilizar para otros espectáculos quedándose así agrupados a la hora de realizar una búsqueda.
R026 Cambios	
Descripción	Todos los campos son modificables a excepción de los identificadores. Se podrán hacer tantos cambios como se necesita por parte de cualquier usuario avanzado o administrador.
R027 Gestión de eventos	
Descripción	Cualquier usuario avanzado o administrador podrá crear, modificar o eliminar eventos.
R028 Fechas de publicación y baja de eventos	
Descripción	Cada evento deberá llevar obligatoriamente las fechas de publicación y vencimiento del mismo, lo que permitirá saber a la aplicación cuándo deberá estar publicado cada uno de ellos. Tanto la fecha de publicación como de baja podrán modificarse en cualquier momento.
R029 Imágenes para eventos	
Descripción	Cada evento podrá tener una imagen asociada que se publicará en el caso de que exista.
R030 Enlaces de eventos	
Descripción	Cada evento podrá tener una url asociada con la que se enlazará cada uno de ellos.

R031	Página de inicio
Descripción	<p>La página de inicio estará dividida en tres bloques:</p> <ul style="list-style-type: none"> - Índice de páginas de información general diferenciado en dos bloques. - El listado de espectáculos de la temporada ordenados cronológicamente a partir de la fecha actual - Listado de eventos actuales, con fecha comprendida entre la fecha de publicación y la de vencimiento.
R032	Página de programación
Descripción	<p>Existirá una página de programación que mostrará todos los espectáculos de la temporada, además permitirá realizar búsquedas de espectáculos por temporada, curso, categoría, género y calificación por edades.</p>
R033	Fichas de espectáculos
Descripción	<p>Todos los espectáculos se mostrarán con el mismo formato. Cada ficha muestra únicamente los campos que hayan sido rellenados.</p>
R034	Páginas de información general
Descripción	<p>Existirán además páginas de información general. Cada bloque de información que se muestre en el menú contendrá además de la información propia de la página, los enlaces a las diferentes subpáginas de ese bloque.</p>
R035	Gestión de páginas
Descripción	<p>Los usuarios avanzados o administrador serán los únicos que podrán insertar, modificar, eliminar y páginas.</p> <p>Para crear nuevas páginas se podrá elegir el menú en el que se quiere que esté la página además de la posición en la que tenga que estar dentro de dicho menú.</p>

R036 Subpáginas	
Descripción	Cada página podrá tener tantas subpáginas como necesite, cada una de ellas tendrá un identificador de su página padre y además se podrá elegir el lugar que ocupará el enlace dentro de la página.

R037 Imágenes en páginas	
Descripción	Cada página podrá tener asociada una imagen.

R038 Control de usuarios	
Descripción	El usuario Administrador es el único que podrá crear o eliminar usuarios avanzados, además también será el único que podrá ver qué cambios ha hecho cada usuario.

R039 Almacenamiento de contraseñas	
Descripción	Las contraseñas serán almacenadas en la base de datos en un formato cifrado.

R040 Inserción de notas y dossiers de prensa	
Descripción	La aplicación tendrá un apartado de notas y dossiers de prensa que permitirán dar a conocer la programación a diferentes medios de comunicación.

3.3. ANÁLISIS DE CASOS DE USO

A partir de los requisitos identificados anteriormente, y con el fin de formalizar éstos, se realiza el Análisis de Casos de Uso mediante diferentes *Diagramas de Casos de Uso*, donde quedan reflejadas todas las acciones que pueden llevar a cabo los usuarios del sistema. En primer lugar, se modela el *Diagrama de Contexto* que representa el contexto del sistema, y proporciona una visión de la situación del mismo a sus usuarios externos (o *actores*) a través de la representación de éstos (pueden ser humanos u otros sistemas), los *casos de uso* y sus correspondientes relaciones; Es decir, un *Diagrama de Casos de Uso* que contiene los diferentes usuarios y sistemas externos que interaccionan con el sistema.

En la siguiente figura se muestra el *Diagrama de Contexto* de la aplicación Auditorio:

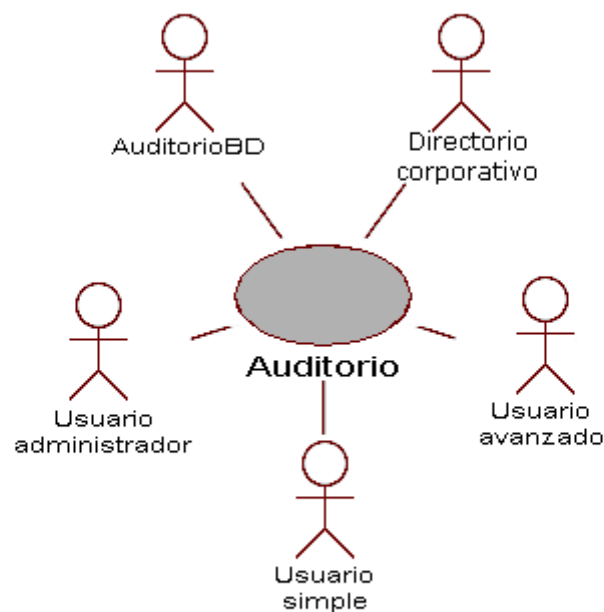


Figura 3.1. Diagrama de Contexto de Auditorio

Se puede apreciar que existen cinco *actores*, entre usuarios y sistemas externos, que interactúan con el sistema, son los siguientes:

- ➡ Administrador, Usuario Avanzado, y Usuario Simple: usuarios que desempeñan los respectivos perfiles comentados con anterioridad.
- ➡ AuditorioBD: sistema que incorpora una base de datos que almacena la información necesaria para llevar a cabo su publicación. Este repositorio recopila, entre otros, los datos personales de los usuarios del sistema (necesarios para poder registrar los usuarios administradores y avanzados), los espectáculos de todas las temporadas, rutas de imágenes, archivos de descargas, notas de prensa, eventos etc.
- ➡ Servicio de Directorio Corporativo: sistema externo de Directorio (LDAP) al cual se realizan consultas para autenticar a todos los usuarios que quieran acceder al sistema, y obtener de esta forma sus datos personales actualizados.

Además del Diagrama de Contexto, a continuación se muestran los Diagramas de Casos de Uso de los diferentes perfiles de usuarios de la aplicación con el fin de observar la funcionalidad que pueden desarrollar cada uno de ellos en el sistema. En primer lugar se presenta el diagrama correspondiente al perfil Usuario Simple:

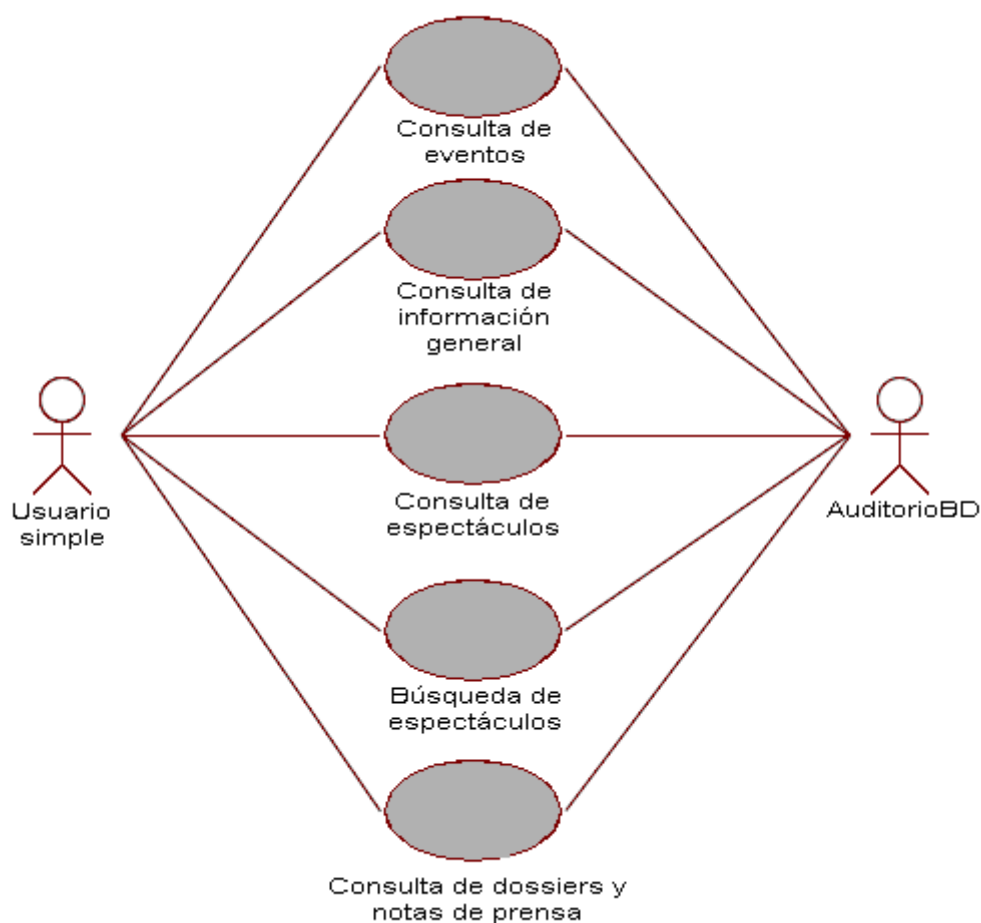


Figura 3.2. Diagrama de Casos de Uso del Usuario Simple

Se puede observar que el perfil Usuario Simple únicamente puede realizar funciones de consulta de la información almacenada en el sistema. Las búsquedas que podrá realizar serán, espectáculos, eventos y noticias además se podrá consultar toda la información almacenada en el sistema navegando por las diferentes páginas.

Por su parte el Diagrama de Casos de Uso del Usuario Avanzado muestra que este perfil de usuario puede realizar tanto funciones de búsqueda como de actualización de los espectáculos, introducción de eventos, campos de búsqueda, notas de empresa o información general. Estas funciones quedan explicitadas en el diagrama que se expone a continuación:

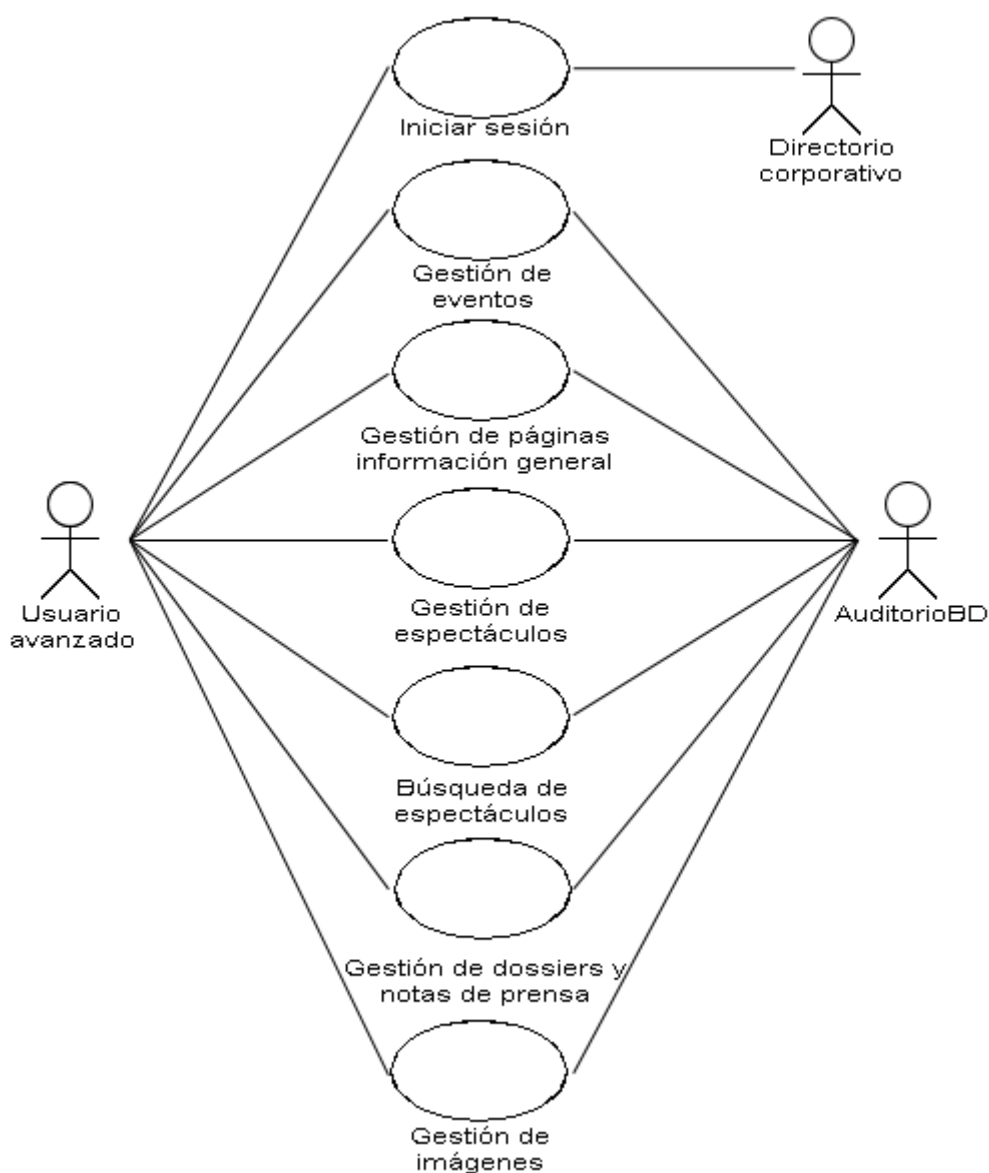


Figura 3.3. Diagrama de Casos de Uso del Usuario Avanzado

El perfil Administrador adquiere mayores responsabilidades en el sistema, pudiendo desempeñar un mayor número de funciones, además de las desarrolladas por el Usuario Avanzado, es el encargado de gestionar los perfiles de usuario. Estas funciones quedan reflejadas en la siguiente figura:

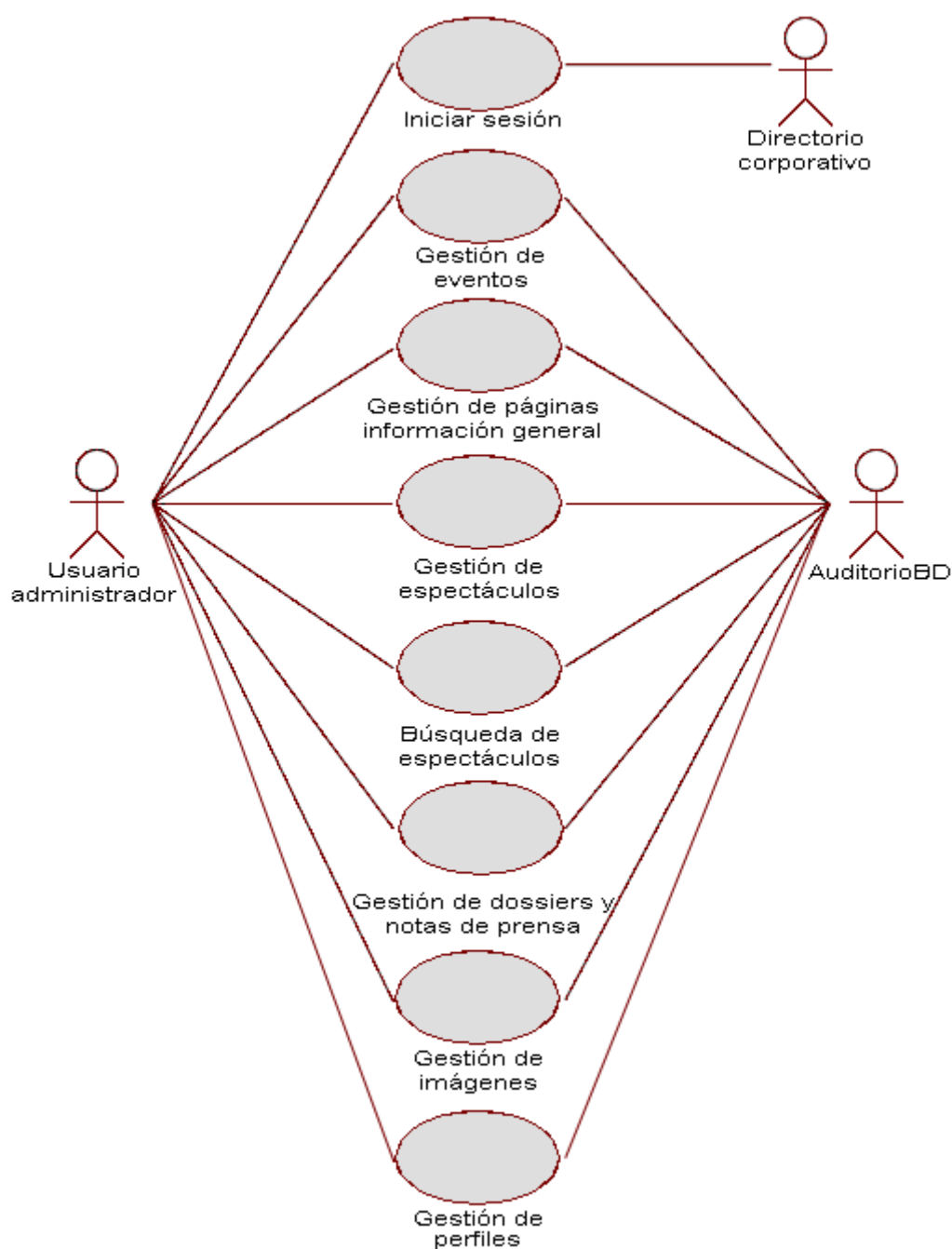


Figura 3.4. Diagrama de Casos de Uso del Administrador

3.4. ANÁLISIS DE CLASES

Teniendo en cuenta las funcionalidades identificadas en el Análisis de Casos de Uso, se elabora un modelo de clases (en fase de análisis), que incluye cada una de las clases surgidas, así como las relaciones entre ellas, con el fin de establecer un modelo de negocio para la aplicación.

Este Diagrama de Clases no trata de ser exhaustivo, por ello no incluye los métodos y atributos de las clases, únicamente trata de establecer en una fase de análisis, las clases y las asociaciones, agregaciones y/o generalizaciones existentes entre ellas.

Las clases que forman parte de este Diagrama de Clases y las relaciones existentes entre ellas se exponen brevemente a continuación:

- Espectáculo: representa la información de cada espectáculo, constituyendo la entidad clave sobre la cual se lleva a cabo toda la gestión de espectáculos. Pueden ser gestionados por los usuarios avanzados o por el administrador. Y pueden ser vistos por los usuarios simples.
- Categoría: entidad que clasificará por categorías los espectáculos. Se consideran categorías las temporadas de espectáculos (otoño, invierno, primavera) o aquellos espectáculos que se engloben dentro de un programa concreto. Pueden ser gestionadas por los usuarios avanzados o por el administrador. El usuario simple podrá usar la búsqueda por categorías para ver los espectáculos que le puedan interesar. Cada espectáculo podrá pertenecer a una o más categorías.
- Calificación: clase que modela las calificaciones por edades de los espectáculos. Tanto el administrador como el usuario avanzado podrán crear, modificar, eliminar calificaciones o cambiar la calificación de un espectáculo por otra usando una ya definida o creando una nueva si esta no existiera. El usuario simple podrá usar la búsqueda por calificaciones para ver los espectáculos definidos para determinados rangos de edad.
- Género: entidad que clasificará los espectáculos por géneros artísticos. Pueden ser gestionadas por los usuarios avanzados o por el administrador. El

usuario simple podrá usar la búsqueda por géneros para ver los espectáculos que le puedan interesar. Cada espectáculo podrá pertenecer únicamente a un género.

- ➔ Compañía: clase que modela compañía que representa un espectáculo. Tanto el administrador como el usuario avanzado podrán crear, modificar, eliminar compañías o cambiar una ya definida y asignada a un espectáculo creando una nueva si ésta no existiera. El usuario simple podrá usar la búsqueda por compañías para ver los espectáculos que le interesan si están interesados en una compañía concreta. Cada espectáculo será representado por una única compañía.
- ➔ Perfil: clase que modelará los diferentes perfiles del público de Auditorio a la hora de asignar precios a las entradas. Los usuarios administrador y avanzado podrán crear, borrar y/o modificar perfiles, mientras que el usuario simple únicamente podrá visualizar aquellos que tengan precio asignado en la tabla de precios propia del espectáculo.
- ➔ Precio: entidad que almacenará todos los precios de un espectáculo. Los precios estarán diferenciados por perfiles de usuarios y por zonas, existiendo tres zonas fijas (zona A, zona B y zona única). Los precios sólo podrán ser modificados por usuarios avanzados o por el administrador y visualizados por el usuario simple en la página de información de cada espectáculo.
- ➔ Fecha: clase que modela las fechas de los espectáculos y de los eventos. Permite ordenar los próximos eventos y espectáculos por orden cronológico para que sea más sencilla su búsqueda.
- ➔ Enlaces de interés: clase que modela los enlaces realizados desde los espectáculos, se permitirá almacenar el enlace para utilizarlo en otros espectáculos o se almacenará sólo para el espectáculo que se esté editando. Los Enlaces de interés podrán gestionarlos los usuarios avanzado y administrador, mientras que el usuario simple únicamente podrá visualizarlo en el espectáculo correspondiente.

- ➔ Imágenes: entidad que almacenará todas las imágenes existentes. Dependiendo de la ubicación de la imagen la aplicación la almacenará en una carpeta diferente con el número de identificador del espectáculo como nombre del archivo. Las imágenes podrán ser gestionadas por los usuarios avanzado y administrador, mientras que el usuario simple únicamente podrá visualizarla en la página correspondiente.
- ➔ Ubicación: clase que modelará las diferentes ubicaciones a las que puede pertenecer una imagen. Esta clase sólo podrá gestionarlas los usuarios avanzado y administrador, siendo su utilización transparente al usuario simple.
- ➔ Descargas: clase que modela todas las descargas existentes en la aplicación. Las podrán gestionar usuarios avanzados y administrador. El usuario simple únicamente las podrá visualizar en el lugar en el que se haya elegido para mostrarlas.
- ➔ Tipo de descarga: cada descarga puede pertenecer a una nota de prensa, un video de un espectáculo o información sobre espectáculos demasiado extensa como para incorporarla a la ficha del espectáculo. Esta entidad almacena los diferentes tipos de descarga para mostrarlos en la página correspondiente. Los tipos de descarga serán gestionados por usuario avanzado y administrador, siendo transparente para el usuario simple.
- ➔ Formato: entidad que distingue los distintos formatos a los que pueden pertenecer la clase descarga, ya que pueden almacenarse pdf, videos, música etc. En el lugar en el que se haya elegido añadir la descarga aparecerá el icono correspondiente al formato del archivo junto a su nombre. Los formatos serán gestionados por usuario avanzado y administrador, siendo transparente para el usuario simple.
- ➔ Prensa: clase que modelará las noticias y notas de prensa publicadas en la aplicación. Ambas podrán ser gestionadas por los usuarios avanzado y administrador, mientras que el usuario simple únicamente podrá visualizarlas. Las notas de prensa estarán referenciadas con un identificador de descarga.

- ➔ Página: clase que modela las páginas web existentes en la aplicación. Se podrán crear, modificar y eliminar páginas. Cada página podrá contener subpáginas, creándose un árbol de directorios. Además, se podrá establecer el orden en el que quieran mostrar los enlaces.
- ➔ Usuarios: clase que almacena los usuarios con permisos para modificar datos de la aplicación. Únicamente el usuario administrador podrá dar de alta/baja o modificar los permisos de los usuarios.
- ➔ Evento: clase que almacena todos los eventos que acontecen en el Auditorio y que se les quiera dar una importancia relevante. Los usuarios simples podrán verlos en la home de Auditorio y sólo el usuario avanzado y el administrador podrá introducir un evento en la aplicación.
- ➔ Gestión de páginas: clase que almacena el registro de creación, modificación o eliminación de páginas en el sistema.
- ➔ Gestión de espectáculos: clase que almacena el registro de creación, modificación o eliminación de espectáculos en el sistema.
- ➔ Gestión de prensa: clase que almacena el registro de creación, modificación o eliminación de noticias en el sistema.

En la figura siguiente se muestra el Diagrama de Clases (de análisis) del sistema:

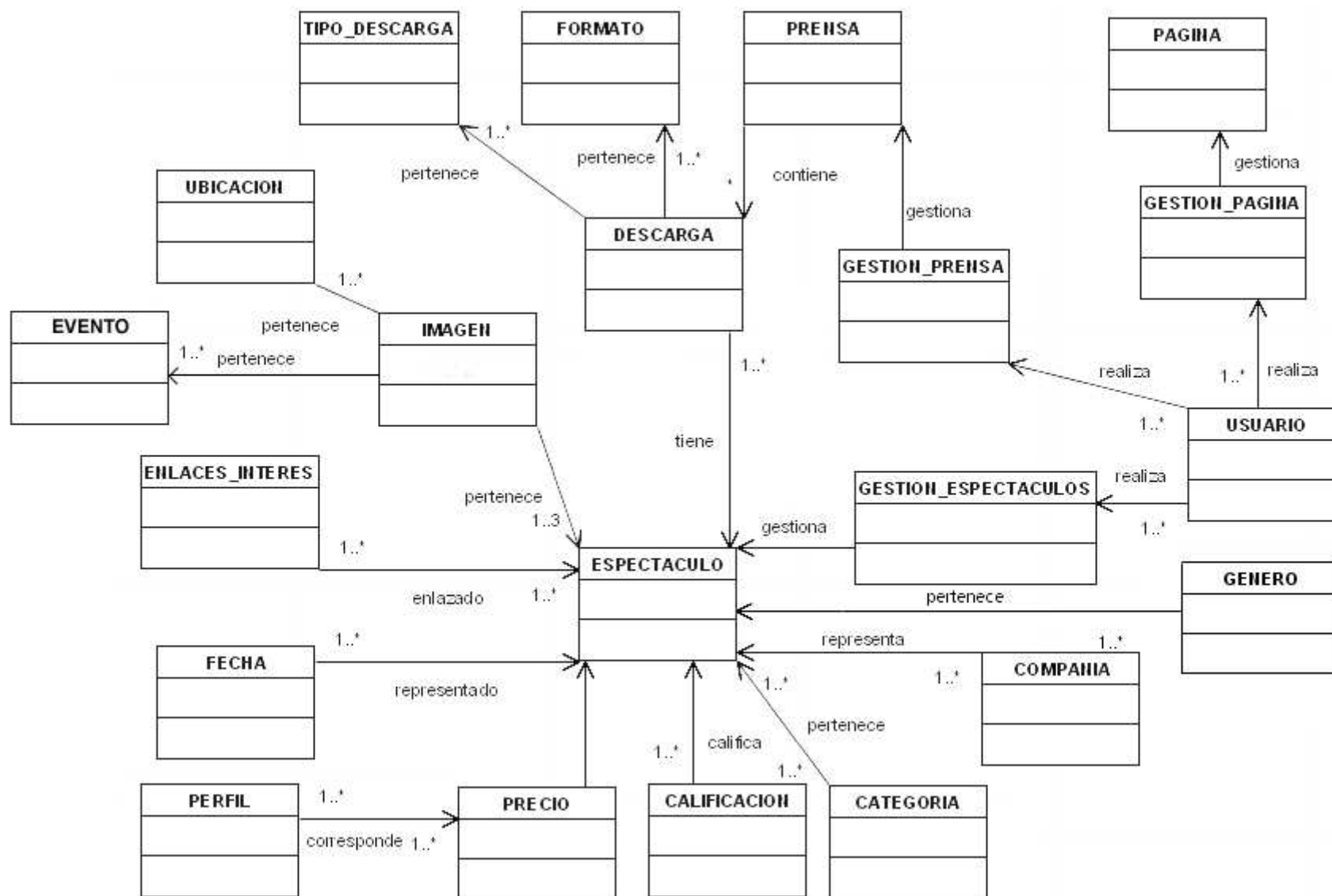


Figura 3.5. Diagrama de clases (Fase de análisis)



CAPÍTULO 4

DISEÑO DEL SISTEMA

El objetivo de este capítulo es obtener el diseño en detalle del sistema de información a través de la definición de la arquitectura del sistema, del entorno tecnológico que le va a dar soporte, y la especificación detallada de los componentes del mismo.

La definición de la arquitectura comprende la partición física del sistema de información, la organización software en subsistemas de diseño y la especificación del entorno tecnológico sobre el que se despliegan dichos subsistemas.

La especificación detallada de los componentes del sistema viene dada por la división de éste en subsistemas de diseño, el Diseño de Clases detallado y el Diseño Físico de Datos.

4.1. DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA

En este apartado se pretende detallar el diseño arquitectónico del sistema. Para ello se proporciona la descripción los subsistemas software que forman parte de la aplicación, y la ubicación de cada subsistema en particiones físicas del sistema, así como la especificación detallada de la infraestructura tecnológica necesaria para dar soporte al sistema de información.

4.1.1. Arquitectura física

En este apartado se especifica la división física del sistema identificando los nodos y las comunicaciones entre nodos. Se entiende por nodo cada partición física o parte significativa del sistema de información con características propias de ejecución y función, e incluso de diseño y construcción.

El particionamiento físico del sistema debe permitir organizar el sistema de información *de forma distribuida*, estableciendo una arquitectura multinivel cliente/servidor, en la cual, los múltiples clientes realizan peticiones al servidor, y el servidor responde a esas solicitudes.

Cliente Web

El cliente en la aplicación constituye una interfaz gráfica de usuario (GUI) correspondiente con el lado cliente del protocolo HTTP (Hypertext Transfer Protocol), es decir, lo que se denomina comúnmente como navegador Web. Este cliente debe permitir además ejecutar algunas tecnologías Web conocidas como “*del lado del cliente*”, como Javascript, CSS (Hojas de estilo en cascada), etc.

Servidor Web

El servidor, constituye el componente HTTP del lado del servidor, comúnmente denominado servidor Web, donde se depositan las páginas Web que solicita el cliente. Este servidor atenderá las peticiones de los clientes y enviará las páginas HTML requeridas a los mismos.

Servidor de Aplicaciones

Además de las páginas HTML, el servidor debe realizar otras funciones fundamentales para el sistema, especialmente debe permitir ejecutar ciertos programas o *scripts*, a petición del cliente, que constituyen, en gran medida, la lógica de negocio de la aplicación. Esta tarea lo convierte en lo que se denomina Servidor de Aplicaciones, que entiende las peticiones de ejecución de los clientes, genera una respuesta dinámica, y la envía al navegador del cliente.

Servidor de Bases de Datos

La generación de estas páginas HTML dinámicas comentadas anteriormente, puede requerir en la mayoría de ocasiones, la consulta a una base de datos, con el fin de satisfacer una petición sobre los datos almacenados. Estos datos deben organizarse y gestionarse a través de un servidor de Bases de Datos, cuya información pueda ser accedida por el servidor de aplicaciones para satisfacer las solicitudes de los clientes.

Servidor de Directorio

Además de los servidores comentados hasta ahora, resulta necesaria la interacción con un servidor de directorio, para la verificación de acceso de los usuarios del sistema, así como para la consulta de sus datos personales.

Con todo esto, la arquitectura física la aplicación, queda reflejada en la siguiente figura:

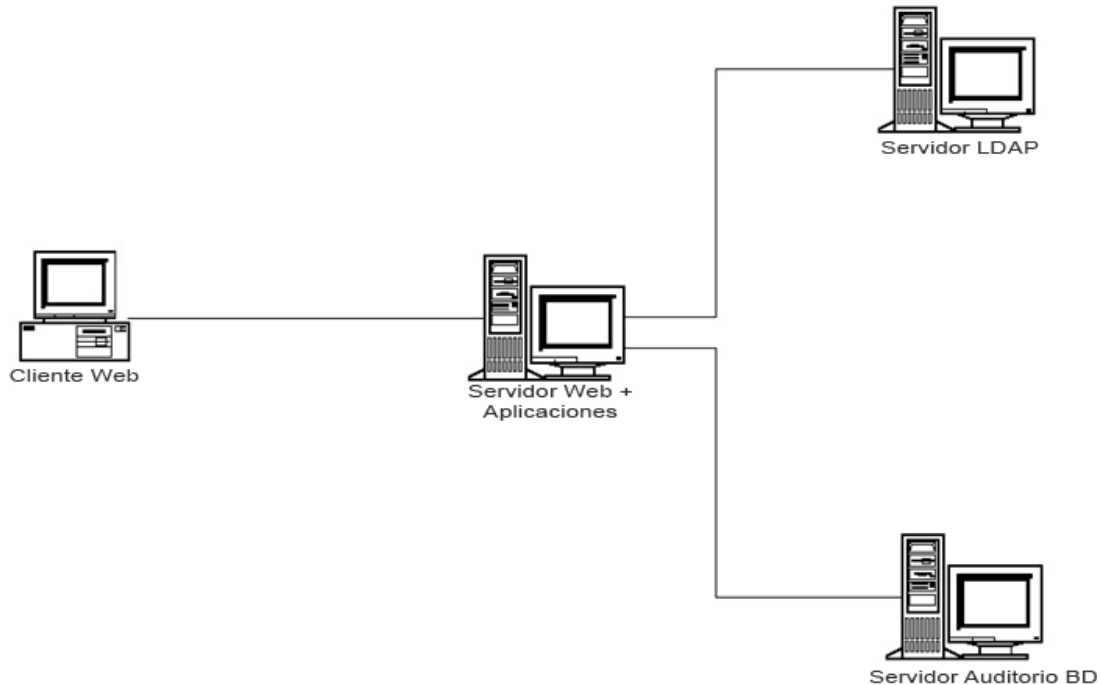


Figura 4.1. Arquitectura Física de Auditorio

4.1.2. Arquitectura lógica

Persigue dividir de forma lógica el sistema de información en subsistemas de diseño, con el fin de reducir la complejidad y facilitar su mantenimiento. La arquitectura software del sistema está basada en la especificación J2EE de Sun Microsystem. Esta especificación define la arquitectura estándar para el desarrollo basado en componentes distribuidos para aplicaciones multicapa. Basándose en la especificación J2EE, se han intentado establecer en el diseño lógico del sistema las siguientes capas: cliente, presentación, negocio, integración, y recursos.

Capa de Cliente

La capa del cliente utiliza una gran variedad de tecnologías para mostrar la GUI al usuario. Se utilizan tecnologías como HTML, DHTML, CSS, y JavaScript del lado del cliente.

Capa de Presentación

Es la encargada de recibir mensajes de la capa cliente y comunicarlos correctamente a la capa de negocio para su procesamiento. La capa de presentación utiliza distintas tecnologías para alcanzar una arquitectura software flexible y sostenible basada en componentes que incluyen Servlets, y Java Server Pages (JSP).

Capa de Negocio

Tiene como finalidad procesar y servir las peticiones de la capa de presentación, proporcionando la lógica de negocio específica de la aplicación y los servicios de control de concurrencia y acceso a la capa de recursos a través de los subsistemas de integración. Este componente se realiza a partir de las clases Java estándar.

Capa de Integración

La capa de integración proporciona la comunicación con el subsistema de recursos. La capa de integración, haciendo uso de la especificación J2EE, proporciona la capacidad de manejar transacciones, seguridad, escalabilidad y el acceso a los recursos. Las distintas tecnologías que se usarán en la capa de integración son JDBC (para el acceso a Base de Datos) y JNDI (para el acceso al directorio corporativo de la universidad).

Capa de Recursos

Tiene como responsabilidad albergar el servicio de directorio y la base de datos de la aplicación.

4.1.3. Relación arquitectura física-lógica

La relación entre los componentes definidos en el diseño arquitectónico lógico y físico permite una especificación de la distribución de los elementos del sistema de información.

En la siguiente figura se muestra el Diagrama de Despliegue de la aplicación con el fin de representar cómo están configuradas las capas lógicas de la arquitectura dentro de la red física identificada en el apartado anterior.

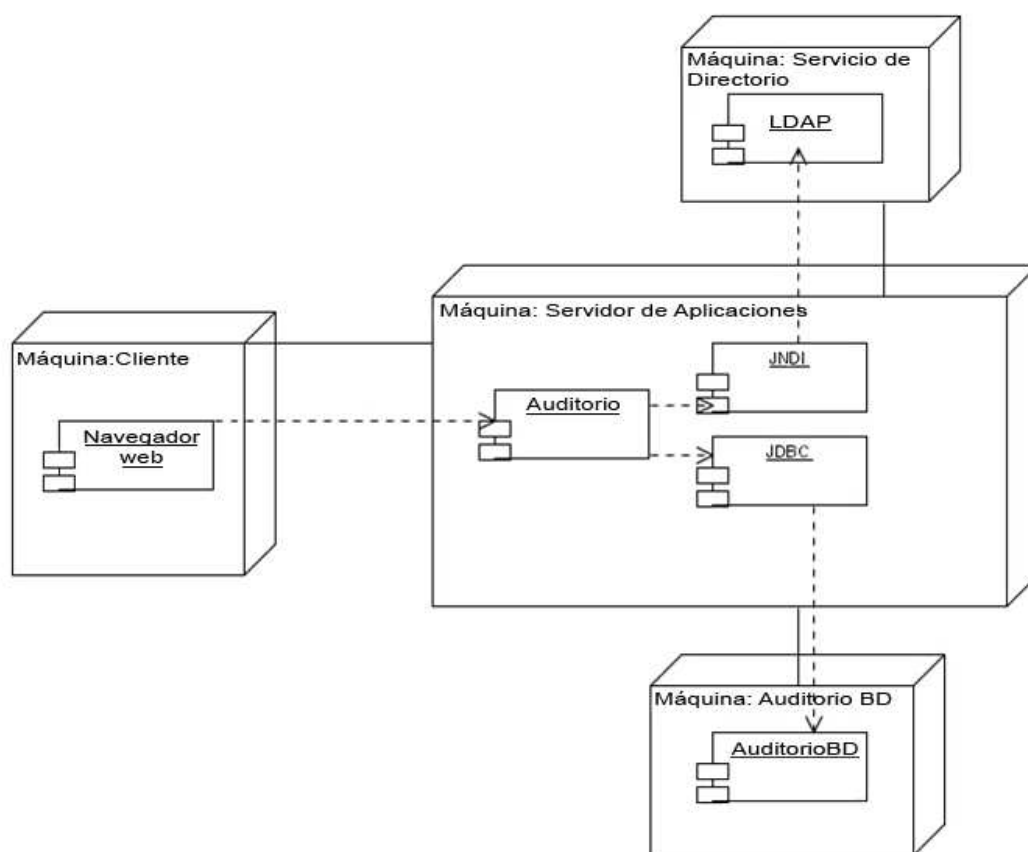


Figura 4.2. Diagrama de Despliegue de Auditorio

El diagrama se compone de cuatros nodos hardware, es decir, cuatro máquinas o elementos físicos donde se encuentran los componentes software del sistema.

Cada elemento físico contiene una configuración de componentes que se ejecuta dentro de la máquina. De esta forma:

El Servidor de Aplicaciones aloja tanto el Servidor Web, como el Servidor de Aplicaciones y todos los elementos software de la aplicación, así como los componentes que permiten la interacción con la Base de Datos y el Servidor de Aplicaciones, JDBC y JNDI respectivamente (capas de *Presentación*, *Negocio* e *Integración*).

El Servidor de Base de Datos contiene el sistema gestor de Base de Datos encargado de administrar y almacenar los datos de la aplicación (capa de *Recursos*).

El Servidor de Directorio ejecuta el Servicio de Directorio creado en la aplicación (capa de *Recursos*).

La/s máquina/s cliente ejecutan un navegador Web que se comunica a través del protocolo HTTP con el Servidor Web (capa de *Cliente*).

Estos componentes software mantienen entre sí una serie de dependencias, expuestas en el diagrama, que reflejan las relaciones existentes entre los diferentes subsistemas localizados en los nodos físicos.

Además, cabe destacar, que todos los elementos físicos incluidos en el diagrama se comunican entre sí a través de la red corporativa de la Universidad, sobre la pila de protocolos TCP/IP.

4.1.4. Elección del entorno tecnológico

En este punto, se define el entorno tecnológico que se requiere para dar respuesta a las necesidades de información, especificando sus posibles condicionantes y descripciones.

Teniendo en cuenta estas necesidades, las tecnologías más adecuadas requeridas para el desarrollo del sistema son discutidas a continuación. La descripción detallada de cada una de ellas puede encontrarse en el Anexo A.

Servidor de Aplicaciones

El servidor de Aplicaciones utilizado debe implementar la especificación J2EE, proporcionando un contenedor que gestione y ejecute los servlets, y contenga un motor de ejecución de las páginas JSP. Existen varios Servidores de Aplicaciones en la actualidad en el mercado, entre los que se encuentran Sun ONE Application Server, Oracle WebLogic Server y **Apache Jakarta Tomcat Server** (cuyas características principales se detallan en el anexo A), que debido a su característica de software libre, y por aportar todas las funcionalidades requeridas para el desarrollo de la aplicación, es seleccionado como el Servidor de Aplicaciones del sistema.

Servidor Web

La elección de Apache Jakarta Tomcat como Servidor de Aplicaciones, puede conllevar también la utilización del Servidor Web incorporado como extensión de Tomcat, o bien la elección de un servidor Web independiente, como puede ser Apache Web Server, Internet Information Server, Sun ONE Web Server, etc. Ya que el servidor Web de Tomcat reúne las características adecuadas para soportar la funcionalidad requerida por la aplicación, se considera una elección idónea para el Servidor Web del sistema.

Base de Datos

Se ha optado por utilizar un Sistema Gestor de Bases de Datos Relacional, por su sencillez, velocidad, flexibilidad, y su organización de los datos en forma de tablas o relaciones.

En el mercado, existen en la actualidad diferentes Sistemas Gestores de Base de Datos Relacionales, como son MySQL, Microsoft SQL Server, Oracle, etc. Si bien tanto Microsoft SQL Server como Oracle son paquetes comerciales, MySQL se caracteriza por ser un sistema de libre distribución. Esta característica no ha sido determinante en la elección del Sistema Gestor de Bases de Datos, ya que la Universidad ya contaba con licencia de estas aplicaciones, por lo que se han tenido en cuenta otras características más influyentes en el contexto del proyecto.

Se ha optado por utilizar Oracle como Sistema Gestor de Bases de Datos Relacional debido principalmente a su flexibilidad, su independencia de la plataforma, así como su gran potencia, que permite la gestión eficiente de grandes cantidades de datos, y el soporte a bases de datos distribuidas. Las características del Servidor de Bases de Datos Oracle son detalladas de forma minuciosa en el Anexo A.

4.1.5. Identificación de los subsistemas

En este apartado se exponen los subsistemas de diseño en los que se ha dividido la aplicación, que consta de los siguientes módulos:

Subsistema de Gestión de Espectáculos que lleva a cabo la gestión de espectáculos (creación de uno nuevo, modificación o eliminado de uno existente).

Subsistema de Gestión de Eventos implementa las funciones de gestión eventos (creación, borrado, historial, modificación, elección de los eventos a mostrar, etc.).

Subsistema de Gestión de Perfiles de Usuario desarrolla la asignación de roles a los usuarios de la aplicación.

Subsistema de Gestión de Noticias implementa las funciones de creación, modificación y borrado de notas de prensa.

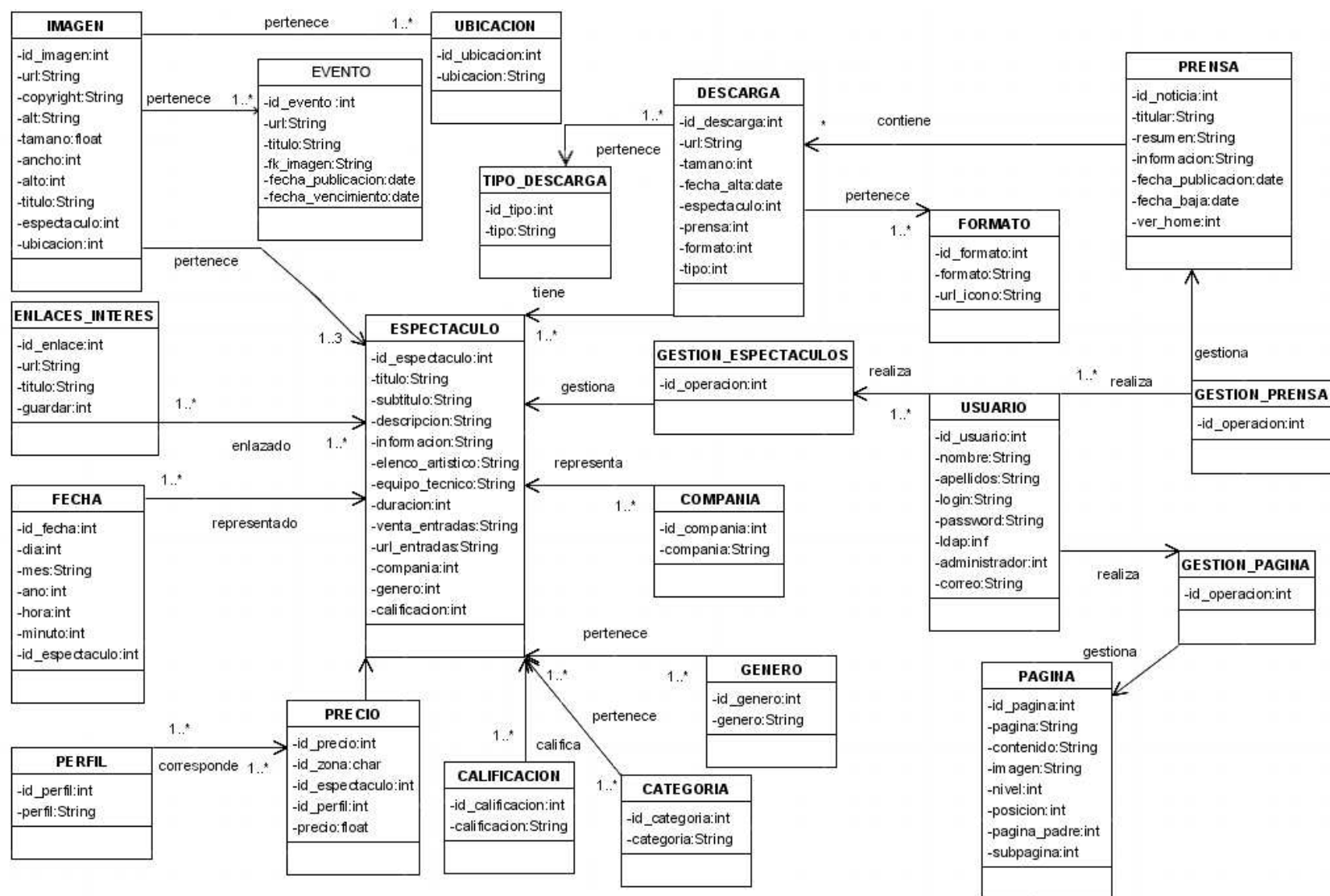
Subsistema de Gestión de Páginas que lleva a cabo la gestión de las páginas web del Auditorio (creación, modificación, organización, borrado, etc.).

Subsistema de Gestión de Descargas lleva a cabo la tarea de inserción, modificación y borrado de descargas asociadas a noticias o espectáculos.



4.2. DISEÑO DE CLASES DETALLADO

Este modelo de clases se elabora en base al modelo de clases de la fase de análisis. Dicho modelo recoge la especificación detallada de cada una de las clases, con sus atributos, y las relaciones establecidas entre ellas. Estas clases corresponden con el modelo de negocio de la aplicación. La siguiente figura lo muestra:



4.3. Diagrama de Clases (Fase de Diseño)

A continuación pasan a detallarse las clases y las relaciones existentes entre ellas:

- Espectáculo: entidad que representa la información de cada espectáculo. Cada espectáculo tiene su propia información y se identifica por un código único, además tiene asignado su título, un subtítulo, la descripción del espectáculo, un resumen de esta descripción para mostrar en la página índice de programación, el elenco artístico, equipo técnico, duración del espectáculo, texto explicativo de cómo se realizará la venta de entradas y la dirección electrónica a través de la cual se podrán adquirir. Además toda la información referente a la compañía, calificación por edades y género quedarán registradas en sus respectivas claves. Esta clase forma parte del subsistema de gestión de espectáculos.
- Categoría: entidad que clasificará por categorías los espectáculos. Se consideran categorías las temporadas de espectáculos (otoño, invierno, primavera) o aquellos espectáculos que se engloben dentro de un programa concreto (por ejemplo Teatro Real). Cada categoría tendrá su propio identificador y nombre. Cada espectáculo podrá pertenecer a una o más categorías. Esta clase forma parte del subsistema de gestión de espectáculos.
- Calificación: clase que modela las calificaciones por edades de los espectáculos. Cada calificación tendrá un identificador propio y su nombre, y estará referenciada desde la clase espectáculos. Esta clase forma parte del subsistema de gestión de espectáculos.
- Género: entidad que clasificará los espectáculos por géneros artísticos. Cada espectáculo podrá pertenecer únicamente a un género y cada uno de éstos tendrá su propio identificador y nombre. Esta clase forma parte del subsistema de gestión de espectáculos.
- Compañía: clase que modela compañía que representa un espectáculo. Cada espectáculo será representado por una única compañía y ésta contendrá un identificador y su nombre. Esta clase forma parte del subsistema de gestión de espectáculos.

- ➔ Perfil: clase que modelará los diferentes perfiles del público del Auditorio a la hora de asignar precios a las entradas formando parte del subsistema de gestión de espectáculos. Cada uno de ellos tendrá su propio identificador y nombre.
- ➔ Precio: entidad que almacenará todos los precios de un espectáculo. Esta clase forma parte del subsistema de gestión de espectáculos y estará identificado por los identificadores de perfil, zona y espectáculo al que pertenece, además almacenará el valor del precio y un identificador del propio precio.
- ➔ Fecha: clase que modela las fechas de los espectáculos y de los eventos. Esta clase forma parte del subsistema de gestión de espectáculos, cada fecha tendrá su propio identificador y almacenará el día, mes, año, hora y minuto propios de la fecha, además tendrá un identificador del espectáculo al que pertenece. La clase se encargará de calcular el día de la semana y el curso al que pertenecen.
- ➔ Enlaces de interés: clase que modela los enlaces realizados desde los espectáculos, se permitirá almacenar el enlace para utilizarlo en otros espectáculos o se almacenará sólo para el espectáculo que se esté editando. Pertenecen al subsistema de gestión de espectáculos y tienen su propio identificador, la url de referencia, el título del enlace y almacenará si se quiere que el enlace se guarde para futuros espectáculos.
- ➔ Imágenes: entidad que almacenará todas las imágenes existentes. Pertenecen al subsistema de gestión de espectáculos y contiene un identificador de imagen además de la ruta en la que está almacenada dentro de la aplicación, su título y texto alternativo, un campo para el copyright si lo tuviera además de campos que almacenaran el alto, ancho y tamaño de la imagen. También tiene una referencia a la ubicación en la que se encuentre la imagen dependiendo del lugar para el que se añada y una referencia al espectáculo al que pertenece.
- ➔ Ubicación: clase que modelará las diferentes ubicaciones a las que puede pertenecer una imagen. Pertenecen al subsistema de gestión de espectáculos y almacena el identificador y el nombre de cada ubicación en la que se pueda

almacenar una imagen. Cada imagen tendrá una referencia de la ubicación que se le haya asignado.

- ➔ Descargas: clase que modela todas las descargas existentes en la aplicación. La clase descarga la comparten los subsistemas de gestión de espectáculos, de gestión de páginas y de gestión de noticias. Cada descarga tendrá un identificador, la ruta en la que se encuentre el archivo dentro de la aplicación, su título, tamaño, y fecha de alta para poder programar con antelación la fecha de publicación del archivo. Además contiene referencias al espectáculo que hace referencia si este es el caso o en su defecto a la noticia o evento. También contendrá referencias al tipo de descarga al que pertenecen así como al formato.
- ➔ Tipo de descarga: esta entidad almacena los diferentes tipos de descarga para mostrarlos en la página correspondiente. Cada una de ellas tendrá un identificador y un nombre y estará referenciado desde la clase descarga.
- ➔ Formato: entidad que distingue los distintos formatos a los que pueden pertenecen la clase descarga, ya que pueden almacenarse pdf, videos, música etc. Cada formato tendrá su identificador, nombre y la ruta en la que esté almacenado el icono al que hace referencia el formato del archivo.
- ➔ Prensa: clase que modelará las notas de prensa publicadas en la aplicación. Pertenece al subsistema de gestión de noticias y contendrá además de su identificador, el titular, el resumen y la información de la noticia, también permitirá almacenar la fecha de publicación y baja y permitirá decidir si se quiere que la noticia aparezca en la página de inicio de Auditorio o en las interiores de noticias y notas de prensa.
- ➔ Página: clase que modela las páginas web existentes en la aplicación. Se podrán crear, modificar y eliminar páginas. Pertenece al subsistema de gestión de página y cada una podrá contener subpáginas, creándose un árbol de directorios. Además, se podrá establecer el orden en el que quieran mostrar los enlaces. Para realizar esta tarea cada clase deberá almacenar además del identificador, el nombre y el contenido, el nivel al que pertenece la página dentro del árbol de

directorios y la posición en la que se colocará en enlace dentro de la página padre. Además si la página no es raíz tendrá la referencia de la página padre y deberá almacenar también si tiene o no subpáginas.

- Usuarios: clase que representa a los usuarios de la aplicación. La información que se almacena de cada usuario es su identificador (login) que es único para cada uno, su nombre y apellidos, su dirección de correo electrónico y si es administrador de la aplicación. Esta clase pertenece al Subsistema de Gestión de Perfiles de Usuario
- Evento: clase que almacena los eventos producidos en el Auditorio, pertenece al subsistema de gestión de eventos. Los eventos se mostrarán en la página de inicio de la aplicación ocupando el lugar del último espectáculo mostrado, si se desea que el texto o la imagen del evento tenga el ancho mayor al de un sector, se deberá especificar el número de sectores a ocupar, quedando menos espacio para espectáculos. Además del número de sectores cada evento almacenará un identificador, el título del evento, la ruta de la imagen si la hubiera y un enlace en el que se amplía la información. También debe contener la fecha de publicación y baja.
- Gestión de páginas: clase que almacena el registro de creación, modificación o eliminación de páginas en el sistema. Para cada actuación aparecerá un código de operación así como la operación realizada, el código de la página implicada y la fecha en la que se produjo la acción.
- Gestión de espectáculos: clase que almacena el registro de creación, modificación o eliminación de espectáculos en el sistema. Para cada actuación aparecerá un código de operación, el código del espectáculo, la operación realizada y la fecha en la que se produjo.
- Gestión de prensa: clase que almacena el registro de creación, modificación o eliminación de noticias en el sistema. Para cada actuación aparecerá un código de operación, el código de la noticia, la operación realizada y la fecha en la que se produjo.

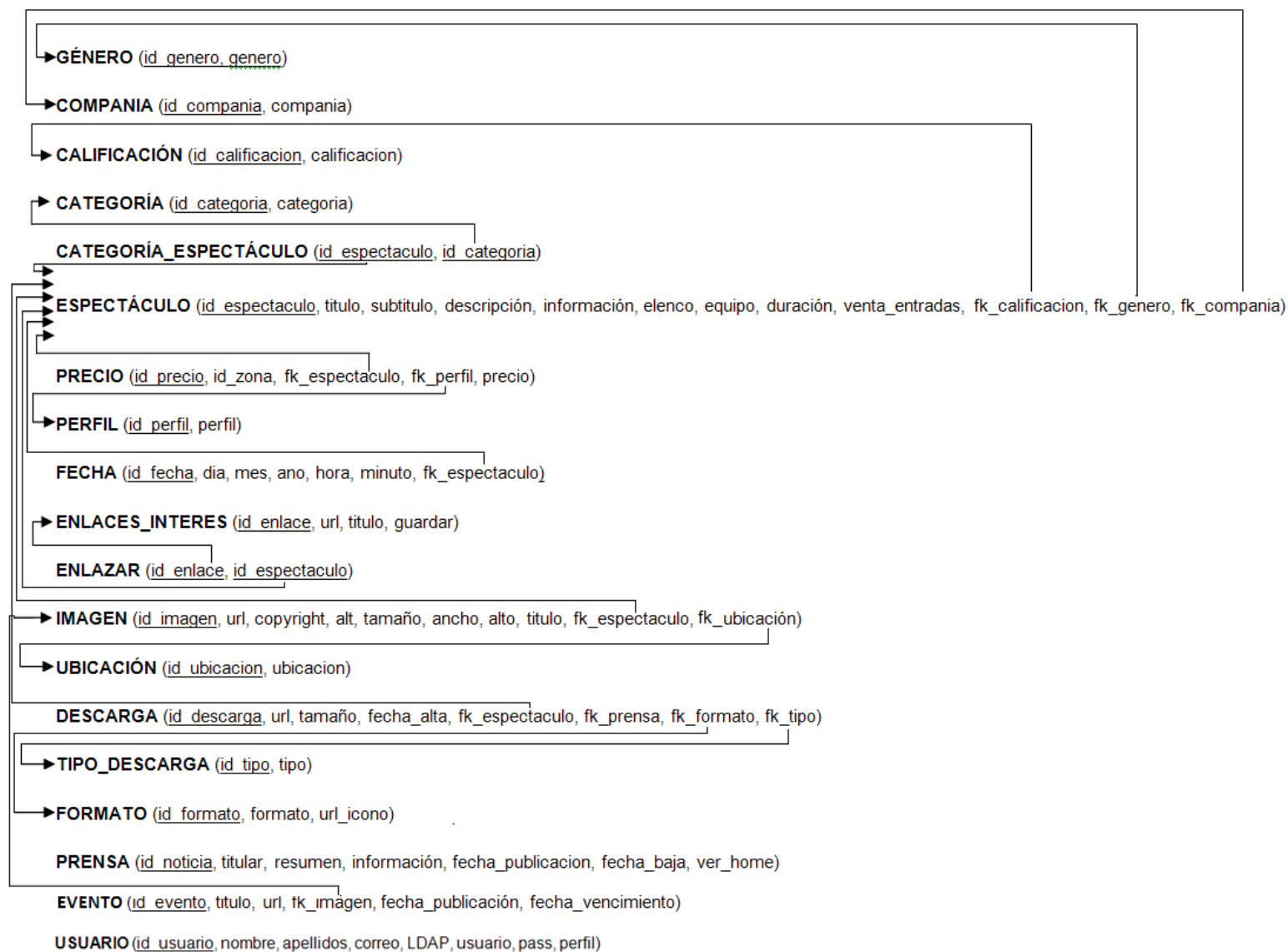
4.3. DISEÑO FÍSICO DE DATOS

En este punto, se define la estructura física de datos que utiliza el sistema a partir del modelo de clases del apartado anterior.

Teniendo en cuenta que la aplicación se desarrolla sobre un sistema Gestor de Bases de Datos Relacional (Oracle), el modelo físico de datos determina cómo se van a convertir las clases en tablas del modelo relacional, considerando las relaciones existentes entre ellas, definiendo sus claves primarias, ajenas, etc.

En él se establecen cada una de las tablas del modelo relacional, y en cada una de ellas las claves primarias (subrayadas) que identifican cada tupla. Las claves ajenas se presentan como referencias (flechas en el diagrama) a las tablas padre a las que hacen referencia.

El modelo relacional resultante para el sistema es el siguiente:





CAPÍTULO 5

Planificación del proyecto y presupuesto

Este capítulo tiene como finalidad realizar un análisis detallado del coste total de la ejecución del presente proyecto, en base al esfuerzo, duración, y recursos necesarios para su realización.

Para calcular este coste, el proyecto ha sido dividido en diferentes tareas que representan las distintas fases de su desarrollo y su evolución en el tiempo. En cada una de estas actividades se determina de forma objetiva la duración y recursos necesarios asociados al desarrollo de cada tarea.

El presupuesto total se obtiene sumando el coste individual asociado a cada tarea con el coste del material utilizado a lo largo de la totalidad del desarrollo del proyecto.

5.1. PLANIFICACIÓN DEL PROYECTO

Un proyecto de desarrollo software consiste en la realización de una serie de actividades prefijadas de antemano teniendo en cuenta la disponibilidad de recursos y restricciones técnicas. Este proyecto se ha dividido en diferentes actividades para su realización.

En algunos casos cada actividad se ha dividido a su vez en tareas. Para cada actividad o tarea se indican los objetivos, las dependencias detectadas, la duración y el trabajo asociado a la misma. El cálculo del trabajo se ha basado en jornada de 8 horas/día.

5.1.1. Actividades

Las actividades principales en las que se ha dividido el proyecto son las siguientes:

- ➔ Estudio de viabilidad.
- ➔ Análisis del sistema.
- ➔ Diseño del sistema.
- ➔ Implementación.
- ➔ Instalación y generación de la memoria integrada del proyecto.

Actividad 1: Estudio de viabilidad

- Descripción: el trabajo durante este período ha consistido en buscar y estudiar posibles soluciones existentes que cubran las necesidades requeridas, evaluando las diferentes alternativas aplicables entre los productos comerciales del mercado y la opción de un desarrollo propio.
- Objetivos:
 - ➔ Búsqueda de soluciones ya implantadas, analizar sus ventajas, inconvenientes y carencias.
 - ➔ Determinar en función del análisis de la información recopilada si está justificado el desarrollo del presente proyecto.

- Estructurar la información recogida y justificar la decisión adoptada. Plasmando esta justificación en el documentación que sirve como introducción de la presente memoria de proyecto.
- Dependencias: esta tarea comienza una vez planteada la realización del proyecto.
- Duración: 10 días.

Actividad 2: Análisis del sistema

Se realiza el proceso de Análisis del Sistema a desarrollar partiendo de los requisitos impuestos por la organización, con el objetivo de obtener una especificación detallada de la aplicación que satisfaga las necesidades de información de los usuarios y sirva como base para el posterior diseño del sistema.

Tarea 2.1. Definición del Sistema

- Descripción: pretende dar una visión general acerca del sistema delimitando su alcance, estableciendo los interfaces con otros sistemas e identificando a los usuarios representativos.
- Objetivos:
 - Determinación del alcance del sistema.
 - Delimitación de los interfaces con otros sistemas existentes.
 - Identificación de los usuarios del sistema y sus funciones.
 - Elección de la metodología, estándares y normas de desarrollo.
- Dependencias: esta tarea comienza al finalizar la actividad 1.
- Duración: 7 días.

Tarea 2.2. Especificación de requisitos

- Descripción: lleva a cabo la definición, análisis y validación de los requisitos a partir de la información facilitada por el usuario, con el fin de obtener un catálogo detallado de los mismos.
- Objetivos:
 - Obtener una especificación detallada de la aplicación que satisfaga las necesidades de los usuarios.
 - Servir como base para el posterior diseño del sistema.
- Dependencias: esta tarea comienza al finalizar la tarea 2.1.
- Duración: 8 días.

Tarea 2.3. Análisis de casos de uso

- Descripción: refleja todas las acciones que pueden llevar a cabo los usuarios del sistema.
- Objetivos:
 - Proporcionar una visión de los usuarios y sistemas externos que interaccionan con el sistema y sus correspondientes relaciones.
- Dependencias: esta tarea comienza al finalizar la tarea 2.1., y puede realizarse paralelamente en el tiempo con la tarea 2.2 según indica la metodología utilizada.
- Duración: 2 días.

Tarea 2.4. Análisis de clases

- Descripción: se elabora un modelo de clases (en fase de análisis), que incluye cada una de las clases surgidas, así como las relaciones entre ellas, con el fin de establecer un modelo de negocio del sistema.
- Objetivos:
 - Establecer en una fase de análisis, las clases y las asociaciones, agregaciones y/o generalizaciones existentes entre ellas.

- Dependencias: esta tarea comienza al finalizar la tarea 2.1., y puede realizarse paralelamente en el tiempo con las tareas 2.2.y 2.3. según indica la metodología utilizada.
- Duración: 5 días.

Actividad 3: Diseño del sistema

El objetivo de esta actividad es obtener el diseño en detalle del sistema de información a través de la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del mismo

Tarea 3.1. Definición de la arquitectura del sistema

- Descripción: detallar el diseño arquitectónico del sistema.
- Objetivos:
 - Descripción los subsistemas software que forman parte de la aplicación.
 - Establecimiento de la ubicación de cada subsistema en particiones físicas.
 - Especificación detallada de la infraestructura tecnológica necesaria para dar soporte al sistema.
- Dependencias: esta tarea comienza al finalizar la actividad 2.
- Duración: 14 días.

Tarea 3.2. Diseño de clases detallado

- Descripción: recoge la especificación detallada de cada una de las clases, es decir, sus atributos, operaciones, métodos y el diseño preciso de las relaciones establecidas entre ellas.
- Objetivos:
 - Transformar el modelo de clases lógico, elaborado en la fase de análisis, en un modelo de clases de diseño.
- Dependencias: esta tarea comienza al finalizar la actividad 2, y puede realizarse paralelamente a la tarea 3.1.

- Duración: 8 días.

Tarea 3.3. Diseño físico de datos

- Descripción: define la estructura física de datos que utiliza el sistema.
- Objetivos:
 - Estructurar los datos en base al Sistema Gestor de Base de Datos a utilizar.
 - Determinar cómo se van a convertir las clases en tablas del modelo relacional.
- Dependencias: esta tarea comienza al finalizar la tarea 3.2.
- Duración: 3 días.

Actividad 4: Implementación del sistema

El método de desarrollo del sistema se basa en una aproximación en espiral, consistiendo cada iteración en la aplicación de un modelo clásico de desarrollo en cascada. El resultado de cada iteración es una versión cada vez más refinada de cada uno de los módulos que componen el sistema.

Tarea 4.1. Versión Inicial

- Descripción: desarrollo inicial del sistema, según las especificaciones iniciales. Siguiendo las fases de desarrollo tradicionales, y partiendo del análisis y diseño del sistema que se ha realizado con anterioridad se realiza un análisis específico de cada uno de los subsistemas, su diseño, implementación y pruebas.
- Objetivos:
 - Definir un interfaz donde quede recogida toda la funcionalidad requerida.
 - Implementar el subsistema de gestión de espectáculos.
 - Implementar el subsistema de gestión de eventos.
 - Implementar el subsistema de gestión de páginas.
 - Implementar el subsistema de gestión de noticias.
- Dependencias: esta tarea comienza al finalizar la actividad 3.

- Duración: 46 días.
- Trabajo: repartido entre las diferentes fases de desarrollo:
 - Análisis: 3 días.
 - Diseño: 8 días.
 - Implementación: 30 días.
 - Pruebas: 5 días.

Tarea 4.2. Primera revisión

- Descripción: una vez que se ha experimentado con una primera versión del sistema, se observó que podía simplificarse el interfaz para facilitar su uso, así como desarrollar la implementación inicial de los subsistemas existentes, así como nuevos módulos necesarios que se incorporaron en esta fase.
- Objetivos:
 - Introducir mejoras en el interfaz y en los subsistemas de la versión inicial.
 - Implementar el subsistema de gestión de usuarios, perfiles y permisos de acceso.
- Dependencias: esta tarea comienza al finalizar la tarea 4.1.
- Duración: 32 días.
- Trabajo: repartido entre las diferentes fases de desarrollo:
 - Análisis: 3 días.
 - Diseño: 6 días.
 - Implementación: 19 días.
 - Pruebas: 4 días.

Tarea 4.3. Segunda revisión

- Descripción: tras obtener un sistema mejorado tras la primera revisión, se percibieron algunas deficiencias mínimas en el interfaz y en los subsistemas desarrollados, así como la introducción de algún módulo nuevo que se incorpora en esta fase.
- Objetivos:
 - Introducir mejoras en el interfaz y en los subsistemas de la segunda versión.
 - Implementar otros subsistemas: gestión de descargas, personalización de interfaz, etc.
- Dependencias: esta tarea comienza al finalizar la tarea 4.2.
- Duración: 12 días.
- Trabajo: repartido entre las diferentes fases de desarrollo:
 - Análisis: 2 días.
 - Diseño: 3 día.
 - Implementación: 5 días.
 - Pruebas: 2 días.

Actividad 5: Instalación y generación de la memoria del proyecto

Tarea 5.1. Instalación de software

- Descripción: instalación de los servidores y el sistema gestor de base de datos, así como de la aplicación y la base de datos asociada.
- Objetivos:
 - Descripción de los pasos de configuración necesarios para la instalación del sistema.
- Dependencias: esta tarea comienza al finalizar la actividad 4.
- Duración: 4 días.

Tarea 5.2. Generación de la documentación

- Descripción: redacción de la memoria final del proyecto.
- Objetivos:
 - ✦ Redacción de la memoria del proyecto fin de carrera.
- Dependencias: esta tarea se ha ido realizando en paralelo a lo largo de todo el desarrollo del proyecto.
- Duración: 25 días.

5.1.2. Resumen del proyecto

En la tabla siguiente se presenta de forma esquemática, todas las actividades y tareas realizadas a lo largo del proyecto, junto con la duración de cada una de ellas, el trabajo realizado y el tiempo empleado.

<i>Actividades</i>	<i>Duración (días)</i>	<i>Total (horas)</i>
1. Estudio de viabilidad		
Analista	10 días	80 horas
	Total Actividad: 80 horas	
2. Análisis del sistema		
2.1. Definición del sistema		
Analista	7 días	56 horas
2.2. Especificación de requisitos		
Analista	8 días	64 horas
2.3. Análisis de casos de uso		
Analista	2 días	16 horas
2.4. Análisis de clases		
Analista	5 días	40 horas
	Total Actividad: 186 horas	
3. Diseño del sistema		
3.1. Definición de la arquitectura del sistema		
Analista	14 días	112 horas



3.2. Diseño de clases detallado		
Analista	8 días	64 horas
3.3. Diseño físico de datos		
Analista	3 días	24 horas
		Total Actividad: 200 horas
4. Implementación del sistema		
4.1. Versión inicial		
Análisis		
Analista	3 días	24 horas
Diseño		
Analista	8 días	64 horas
Implementación		
Programador	30 días	240 horas
Pruebas		
Programador	5 días	40 horas
4.2. Primera revisión		
Análisis		
Analista	3 días	24 horas
Diseño		
Analista	6 días	48 horas
Implementación		
Programador	19 días	152 horas
Pruebas		
Programador	4 días	32 horas
4.3. Segunda revisión		
Análisis		
Analista	2 días	16 horas
Diseño		
Analista	3 días	24 horas
Implementación		
Programador	5 días	40 horas
Pruebas		



Programador	2 días	16 horas
	Total Actividad: 720 horas	
5. Instalación y generación de la memoria del proyecto		
5.1. Instalación del software		
Programador	4 días	32 horas
5.2. Generación de la documentación		
Analista	25 días	200 horas
	Total Actividad: 232 horas	
TOTAL PROYECTO 1418 horas		

5.1.3. Diagrama de Gantt

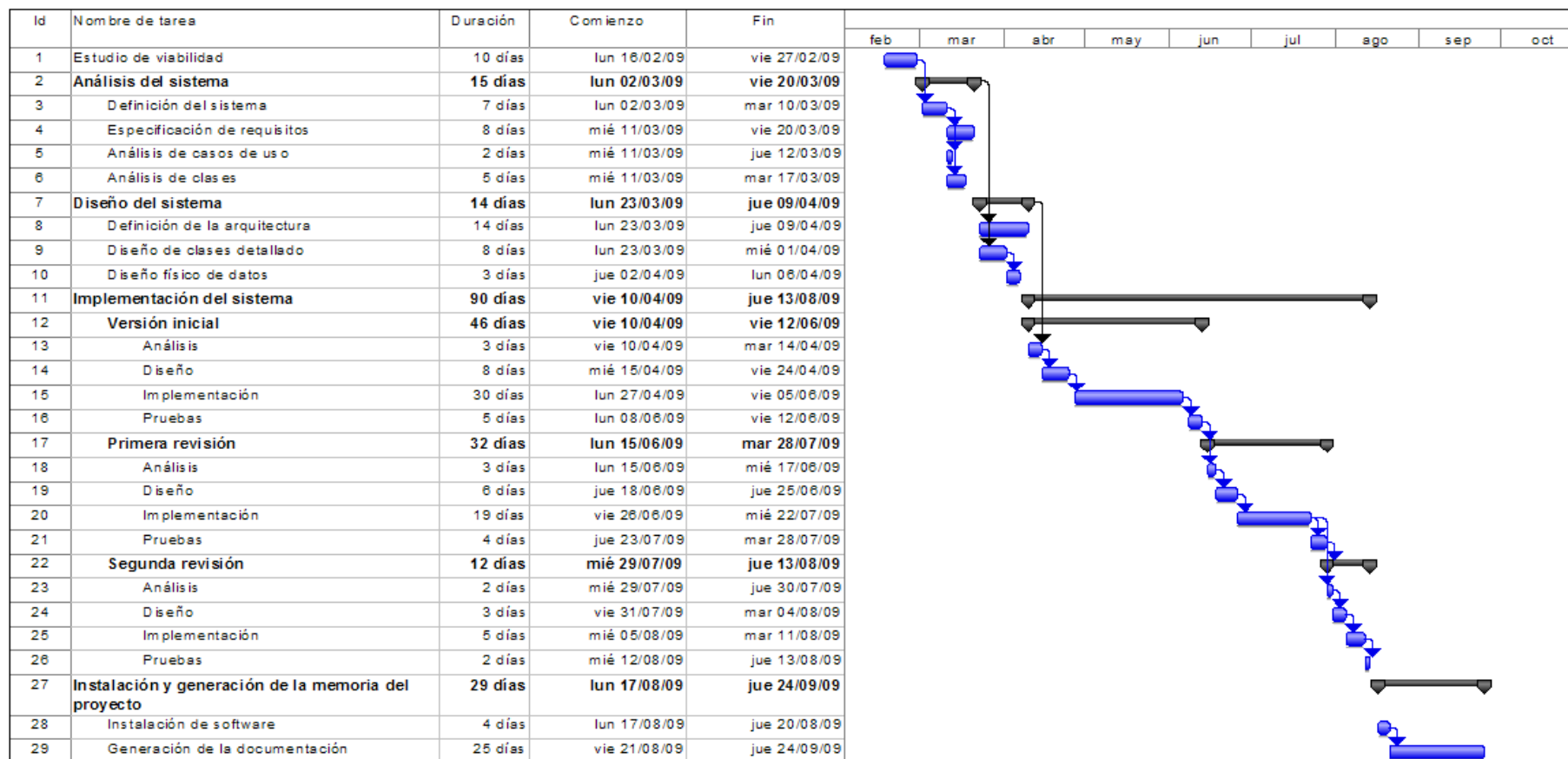


Figura 5.1. Diagrama de Gantt (planificación del proyecto)

5.2. VALORACIÓN ECONÓMICA DEL PROYECTO

En la gestión de un proyecto software resulta necesario realizar cálculos del esfuerzo a realizar en su desarrollo, y sobre los cuales poder estimar el número de recursos necesarios para su correcta consecución, y con ello obtener el coste económico final del proyecto a desarrollar. Por lo que el coste real del desarrollo de este proyecto, es obtenido a partir de los costes del personal y del material utilizado. A continuación se detallan cada uno de estos costes:

Costes de Personal

<i>Concepto</i>	<i>Cantidad (horas)</i>	<i>Coste unitario (euros/hora)</i>	<i>Importe total</i>
Ingeniero Informático	856 horas	25 euros/hora	21.400 euros
Programador	525 horas	20 euros/hora	11.040 euros
Coste total del personal			32.440 euros

Costes de Material

<i>Concepto</i>	<i>Cantidad (unidades)</i>	<i>Coste unitario (euros/unid.)</i>	<i>Importe total</i>
Servidor	1	3563'314 euros	3563'31 euros
Coste total del material			3563,31 euros

Coste total del proyecto

<i>Concepto</i>	<i>Importe</i>
Costes de personal	32.440 euros
Costes de material	3563,314 euros
I.V.A. (16%)	570,13 euros
COSTE TOTAL DEL PROYECTO	36.573,44¹ euros

¹ Véase que se ha realizado una estimación de los costes de personal.

CAPÍTULO 6

Conclusiones y futuras líneas de desarrollo

En este capítulo se revisan los objetivos propuestos en el primer capítulo para validar si se han alcanzado las metas establecidas.

De igual forma, se presentan propuestas y futuras mejoras que pueden realizarse sobre el sistema, y que pueden sentar las bases para próximos proyectos.

6.1. CONCLUSIONES

Los objetivos que se han cubierto con el trabajo realizado son los siguientes:

- ➔ Se ha conseguido desarrollar una herramienta que permite gestionar los espectáculos, eventos, descargas, noticias y páginas de información de cualquier espacio escénico.
- ➔ Se ha logrado realizar búsquedas de espectáculos pasados o futuros a partir de fechas, categorías, géneros, calificación por edades y compañías.
- ➔ Se ha conseguido desarrollar un interfaz que muestre los próximos espectáculos y toda la información relativa a un teatro o auditorio.

Por otra parte está por ver si se alcanzan los objetivos de mejorar la calidad del servicio prestado al cliente, facilitando la gestión de espectáculos y demás información. La consecución de estos objetivos solo podrá constatarse tras la utilización del sistema.

También, merece una especial mención el trabajo realizado para dar de alta espectáculos nuevos y pasados, con el fin de dotar de una mayor riqueza la aplicación. La actualización de la información requerirá de la intervención constante del personal que deberá ocuparse de mantener la programación actualizada.

A título personal, la realización de este proyecto ha contribuido en gran medida a mi formación, tanto por la diversidad de tecnologías y plataformas distintas que he utilizado, como por la libertad a la hora de tomar decisiones de diseño e implementación siempre siguiendo las especificaciones del proyecto. Si bien, en muchas ocasiones, me ha resultado difícil la toma de decisiones y ha demorado los tiempos de desarrollo del proyecto, me ha hecho participar más activamente en el análisis y diseño del sistema no quedando relegada mi posición a la de mero programador.

6.2. FUTURAS LÍNEAS DE DESARROLLO

Las propuestas que quedan pendientes para los trabajos futuros son las siguientes:

- ➔ Realizar una galería de imágenes. La aplicación sólo permite mostrar una imagen de cada espectáculo, lo que puede dar una idea pobre del mismo. La galería permitiría ampliar el número de imágenes mejorando la percepción del espectáculo y así atraer más público al Auditorio.
- ➔ Ampliar los campos de búsqueda permitiendo filtrar espectáculos por perfiles de precios. De esta manera se podría obtener la programación únicamente con el precio que interese al espectador o en un determinado rango de precios.
- ➔ Implementar un subsistema de foros de discusión. Esto permitirá establecer una comunicación entre los espectadores de un acto comentándolo y dando su opinión. Las opiniones reflejadas en el foro quedarán almacenadas y se mostrará en la ficha de cada espectáculo como reclamo de futuros espectadores.
- ➔ Realizar un Buzón de Sugerencias. A través de este buzón, se podrán conocer los intereses del público a la hora de realizar la próxima programación así como su grado de satisfacción con los servicios prestados.
- ➔ Implementar una herramienta que permita al público la subscripción a boletines digitales. Almacenará los datos de los suscritos en la base de datos de la aplicación y permitirá mandar el boletín mensual a todos los usuarios registrados.
- ➔ Adaptar el interfaz a cada usuario permitiendo personalizar las funcionalidades que se presentan, los colores de la ventana, el lenguaje, fuente, etc.

CAPÍTULO 7

BIBLIOGRAFÍA

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *El lenguaje unificado de modelado*. Madrid: Addison Wesley, 1999.

TRIGOS GARCÍA, Esteban; *JSP Guía práctica para usuarios*; Madrid. Ediciones Anaya Multimedia; 2001.

Ceballos Sierra, Francisco Javier; *Java 2: lenguaje y aplicaciones*; RA-MA; 2006.

Ceballos Sierra, Francisco Javier; *Java 2: interfaces gráficas y aplicaciones para internet*; RA-MA; 2008.

Pérez López, César; *Oracle 10g: administración y análisis de bases de datos*; RA-MA; 2005.

Mukhar, Kevin; *Bases de datos con Java: fundamentos*; Anaya Multimedia; 2002.

Xavier Colomés Fornós; Anaya Multimedia; 2008

RECURSOS ELECTRÓNICOS

Tutorial UML, Elementos Comunes a Todos los Diagramas

<<http://www.clikear.com/manuales/uml/elementoscomunes.aspx>>

Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo

<<http://www.pst.informatik.uni-muenchen.de/personen/koch/ideas03-escalona-koch.pdf>>

Diseño del Sistema de Información

<<http://www.csi.map.es/csi/metrica3/dsiproc.pdf>>

Servidores IBM

http://www-05.ibm.com/es/express/?ca=swiotes_es-20090904&me=psearch&met=Q3_es_sx_intel_key_wd_es&re=Adwords&s_tact=&cm_mmc=swiotes_es-20090904-_-_-k-Adwords-_-Q3_es_sx_intel_key_wd_es

Servidores de aplicación

[<http://es.wikipedia.org/wiki/Servidor_de_aplicaciones>](http://es.wikipedia.org/wiki/Servidor_de_aplicaciones)

Foro sobre hojas de estilo CSS

[<http://www.webestilo.com/foros/mensaje.phtml?foro=16&msg=43359>](http://www.webestilo.com/foros/mensaje.phtml?foro=16&msg=43359)

La web del programador. Hoja de estilo

[<http://www.lawebdelprogramador.com/news/mostrar_new.php?id=40&texto=HTML&n1=460588&n2=1&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0>](http://www.lawebdelprogramador.com/news/mostrar_new.php?id=40&texto=HTML&n1=460588&n2=1&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0)

Libros Web. Hojas de estilo

[<http://www.librosweb.es/css/capitulo5/posicionamiento_float.html>](http://www.librosweb.es/css/capitulo5/posicionamiento_float.html)

MÉTRICA versión 3

[< http://www.csi.map.es/csi/metrica3/index.html>](http://www.csi.map.es/csi/metrica3/index.html)

ANEXO A

TECNOLOGÍAS EMPLEADAS

A.1. PLATAFORMA JAVA

Una plataforma es el entorno de hardware o software en el cual se ejecuta un programa.

A diferencia de las plataformas tradicionales como Linux, Windows, y Solaris, la plataforma Java está basada sólo en software que se ejecuta sobre otras plataformas basadas en hardware. Por esto la plataforma Java y su software pueden ser utilizados sobre variados sistemas operativos y hardware.

La plataforma Java está constituida por tres componentes:

- El lenguaje
- La máquina virtual
- Las bibliotecas

Lenguaje Java

El lenguaje Java es un lenguaje de propósito general, de alto nivel, que utiliza el paradigma de orientación a objetos. Su sintaxis y tipos están basados principalmente en C++ ver [URL 05], sin embargo, las diferencias principales con éste son la administración de memoria, siendo ejecutada por la máquina virtual automáticamente y no por el código de cada programa, que ayuda a controlar la sincronización de procesos paralelos. Estas características dan al lenguaje Java las propiedades de robustez y seguridad, evitando por ejemplo problemas de *buffer overflow* utilizados en ataques a sistemas.

Máquina Virtual

Los programas escritos en Java son compilados como archivos ejecutables de una máquina virtual llamada Máquina Virtual Java (*Java Virtual Machine: JVM*). Existen implementaciones de esta máquina para múltiples plataformas, permitiendo ejecutar en diferentes arquitecturas el mismo programa ya compilado. La característica de independencia de la plataforma hace posible el libre intercambio de software desarrollado en Java sin necesidad de modificaciones, lo que es llamado “Escribe una vez y Ejecuta en todas Partes” (*“Write Once, Run Anywhere”*: WORA).

Java es un lenguaje compilado e interpretado a la vez. Compilado ya que previo a su ejecución *bytecodes*. Interpretado porque cada programa luego debe ser procesado y ejecutado por alguna implementación de la JVM específica a la plataforma.

Bibliotecas

El conjunto de bibliotecas del lenguaje es conocido como la Interfaz de Programación de Aplicaciones Java (*Java Application Program Interface: Java API*) que es un gran conjunto de componentes que proporcionan diferentes herramientas para el desarrollo de programas Java. La API de Java está agrupada en conjuntos de bibliotecas relacionadas conocidas como paquetes, que contienen grupos de elementos básicos de java, llamados clases e interfaces.

Ediciones de Java

La plataforma Java ha sido dividida en tres ediciones distintas según sus diferentes objetivos: **Java 2 Micro Edition**, orientado al desarrollo para artículos pequeños y móviles como Dispositivos Asistentes Personales (*Personal Device Assistant: PDA*), **Java 2 Standard Edition**, orientado al desarrollo para ordenadores personales y aplicaciones en general, y **Java 2 Enterprise Edition** orientado al desarrollo de aplicaciones corporativas.

A.1.1. JAVA 2 STANDARD EDITION

Java 2, Standard Edition (J2SE) es la edición principal de la plataforma Java sobre la cual se basan las demás ediciones. Provee las capacidades de desarrollo y ejecución de software escrito en lenguaje Java. Está constituido por dos módulos principales:

- **Kit de Desarrollo de Software J2SE** (J2SE SDK), conocido inicialmente como Kit de Desarrollo Java (*Java Development Kit*: JDK), proporciona el software necesario para desarrollar programas en Java como es el compilador, el depurador y las bibliotecas con las funcionalidades del lenguaje.
- **Entorno de Ejecución Java** (*Java Runtime Enviroment*: JRE), contiene sólo el entorno necesario y las bibliotecas principales para ejecutar software escrito en Java.

J2SE incluye herramientas y APIS para desarrollar aplicaciones con interfaz gráfica, acceso a bases de datos, acceso a directorios, seguridad, entrada/salida, programación en red y varias otras funcionalidades.

Las bibliotecas principales son clases que se encuentran dentro de los paquetes *java*, y las bibliotecas con extensiones estándares se encuentran como clases dentro de los paquetes *javax*. Para poder nombrar bibliotecas de aplicaciones desarrolladas por terceros, se utiliza una convención de invertir el nombre de dominio de Internet del desarrollador y luego separar por funcionalidades, así existe por ejemplo *org.w3c*. y *org.apache*. entre otros.

A.1.2. JAVA 2 ENTERPRISE EDITION

Java 2, Enterprise Edition (J2EE) es una especificación que define una plataforma para crear aplicaciones empresariales utilizando un modelo multicapa, dividiendo la aplicación en diferentes niveles, cada uno especializándose en una tarea en particular. Su estructura está basada en J2SE y un conjunto de sus APIs, a la cual J2EE aporta la especificación de componentes, contenedores y las APIs para los servicios de transacciones, mensajería, envío de correo y conectores de recursos externos.

La especificación de J2EE se encuentra en la versión 1.3, publicada en Agosto de 2001 y está compuesta por un conjunto de paquetes que, si bien algunos con

opcionales para J2SE, son requeridos para la plataforma J2EE. En el momento de elaborar este trabajo aún se trabaja en la elaboración del documento que detallará la versión final de J2EE 1.4.

A.2. ARQUITECTURA J2EE

La especificación de J2EE define su arquitectura basándose en los conceptos de capas, contenedores, componentes, servicios y las características de cada uno de estos. Las aplicaciones J2EE son divididas en cuatro capas: la capa cliente, la capa de presentación o Web, la capa de negocio y la capa de datos, ver Figura:

Capa Cliente

Representada en la figura como Presentación del lado Cliente (Client-Side Presentation) corresponde a lo que se encuentra en el cliente. Es la interfaz gráfica del sistema y se encarga de interactuar con el usuario. J2EE tiene soporte para diferentes tipos de clientes incluyendo clientes HTML, applets Java y aplicaciones Java.

Capa Web

Representada en la figura como Presentación del lado Servidor (Server-Side Presentation) corresponde a lo que se encuentra en el servidor Web. Contiene la lógica de presentación que se utiliza para generar las respuestas a los clientes.

Recibe los datos del usuario desde la capa cliente y basándose en estos datos genera una respuesta apropiada a la solicitud. J2EE utiliza en esta capa los componentes *Java Servlets* y *Java Server Pages*, para crear los datos que se enviarán al cliente.

Capa Negocio

Representada en la figura como Lógica de Negocio del lado servidor (Server-side Business Logic) corresponde a lo que se encuentra en el servidor de aplicaciones.

Contiene el núcleo de la lógica de negocio de la aplicación. Provee las interfaces necesarias para utilizar el servicio de componentes del negocio. Los componentes de negocio interactúan con la capa de datos y son típicamente implementados como componentes *JavaBean* de Empresa (*Enterprise JavaBean*: EJB).

Capa de Datos

Representada en la figura como Sistema de Información Corporativo (*Enterprise Information System*: EIS) corresponde a los recursos de información.

Incluye base de datos, sistemas de procesamiento de datos, sistemas heredados, que deben mantenerse por su importancia y/o complejidad, y sistemas de planificación de recursos.

Esta capa es el punto donde las aplicaciones J2EE se integran con otros sistemas no J2EE o con sistemas heredados.

A.3. APACHE JAKARTA TOMCAT

A.3.1. INTRODUCCIÓN

El servidor Web utilizado para dar soporte a los Servlets de Java es Apache Tomcat. Tomcat es un *contenedor de Servlets* con un entorno JSP. Un contenedor de Servlets es una *shell* en tiempo de ejecución que se encarga de gestionar e invocar los Servlets dependiendo del comportamiento de los usuarios. Puede ser utilizado como un servidor independiente o como una extensión de un servidor Web ya existente como puede ser Apache. Estas son algunas características del servidor Web Tomcat.

A.3.2. ESTRUCTURA DE DIRECTORIOS DE TOMCAT

Después de instalar el servidor de la manera explicada en anteriores apartados, tendremos la siguiente estructura de directorios:

- **bin**
Contiene los *scripts* para arrancar y parar el servidor.
- **conf**
Contiene algunos ficheros de configuración, como `server.xml`, que es el fichero con la configuración principal y `web.xml` que contiene los valores por defecto para las distintas aplicaciones Web desarrolladas en Tomcat.
- **doc**
Contiene distintos documentos que forman la documentación de Tomcat.
- **lib**
Contiene varios archivos *jar* utilizados por Tomcat. En UNIX los ficheros de este directorio se añaden al *classpath* de Tomcat.
- **logs**
En este directorio es donde Tomcat genera sus ficheros log.
- **src**
Aquí se encuentran los códigos fuente de la API de los Servlets, aunque sólo están las interfaces vacías y las clases abstractas que deben ser implementadas por cualquier Servlet container.
- **webapps**
Aquí se encuentran las aplicaciones Web. Por defecto tenemos una serie de aplicaciones de ejemplo.

Además de estos directorios, se pueden crear, o Tomcat puede crear automáticamente, otros dos directorios:

- **work**

Este directorio es generado automáticamente por Tomcat. Aquí es donde el servidor coloca los ficheros intermedios (como ficheros JSP compilados) que genera durante su trabajo. Si se borra este directorio mientras el servidor está arrancando no se podrán, por ejemplo, ejecutar páginas JSP.

- **classes**

Se puede crear este directorio para añadir clases adicionales al *classpath*. Cualquier clase que se ponga en este directorio será añadida al *classpath* de Tomcat.

A.3.3. LOS SCRIPTS DE TOMCAT

Tomcat es un programa Java, en consecuencia es posible ejecutarlo desde la línea de comandos, después de establecer algunas variables de entorno. Sin embargo, establecer cada variable de entorno manualmente y poner los parámetros necesarios en la línea de comandos es una tarea tediosa y propensa a errores. Para evitar esto, el equipo de desarrollo de Tomcat proporciona unos *scripts* para facilitar el arranque y parada de Tomcat.

- **tomcat** : es el script principal. Establece el entorno adecuado, incluyendo *CLASSPATH*, *TOMCATHOME* y *JAVAHOME*, arranca Tomcat con los parámetros adecuados.
- **Startup**: arranca Tomcat en *background*.

A.3.4. FICHEROS DE CONFIGURACIÓN DE TOMCAT

La configuración de Tomcat está basada en dos ficheros:

- **Server.xml:** Es el fichero de configuración principal. Proporciona la configuración inicial para los componentes de Tomcat y permite que el servidor arranque e instancia los componentes que en este archivo se detallan. Para dar de alta una aplicación deberá incluirse en este fichero según se especifica en el apartado de instalación.
- **Web.xml:** En este fichero se almacena la estructura de las aplicaciones Web del servidor. Al arrancar, Tomcat utiliza el fichero por defecto que se encuentra en el directorio *conf*, como configuración base, y el fichero específico de cada aplicación, que se encuentra en el directorio *WEB-INF/*, para sobrescribir los valores por defecto. Se puede renombrar los Servlet desarrollados añadiendo una etiqueta de tipo *Servlet* a este fichero, pero en nuestro caso bastará con los nombres por defecto, que coinciden con el nombre de la clase.

A.4. ORACLE

A.4.1. INTRODUCCIÓN

Oracle es un Sistema de Gestión de Bases de Datos Relacionales que proporciona un enfoque abierto, comprensible e integrado para la gestión de información. Un servidor Oracle consta de una Base de Datos Oracle y de una instancia del Servidor.

Una Base de Datos Oracle consta de una estructura lógica y una estructura física. Como se trata de dos estructuras separadas, el almacenamiento puede gestionarse de forma independiente. La estructura física viene determinada por los archivos del sistema operativo que forman la base de datos, y la estructura lógica viene determinada por áreas lógicas de almacenamiento o *Tablespaces*, y por objetos del esquema de la Base de Datos, estructuras lógicas que hacen referencia directa a los datos de la Base de Datos, como tablas, vistas, procedimientos, etc.

Cada vez que se inicia una Base de Datos, se reserva espacio para un área global del sistema (SGA, *System Global Area*) y se inician los procesos de background de

Oracle. El área global del sistema es un área de memoria utilizada para guardar la información de la base de datos compartida por los usuarios de la Base de Datos. A la combinación de procesos de background y de buffers de memoria se le domina *instancia* Oracle.

Una instancia Oracle tiene dos tipos de procesos: procesos de usuario y procesos de Oracle.

- Un proceso de usuario ejecuta el código de un programa de aplicación.
- Los procesos Oracle, procesos de servidor, que realizan el trabajo necesario para responder a los procesos de usuario, encargados también de realizar tareas de mantenimiento del Servidor Oracle.

A.4.2. ESTRUCTURA DE LA BASE DE DATOS

Una Base de Datos Oracle es una colección de datos que es tratada como una unidad. El propósito de una Base de Datos es permitir guardar y recuperar información. La Base de Datos Oracle cuenta con estructuras lógicas y físicas independientes:

Estructura Lógica de la Base de Datos

Una Base de Datos está dividida en unidades lógicas de almacenamiento denominadas *Tablespaces*. Un tablespace se utiliza para agrupar estructuras lógicas relacionadas. Así, una Base de Datos está dividida en uno o más tablespaces, y para cada tablespace existen uno o más archivos de datos, de forma que se almacenen físicamente las estructuras lógicas.

Un esquema, por su parte, es una colección de objetos. Los objetos son estructuras lógicas que hacen referencia directamente a los datos de la Base de Datos. No hay relación directa entre un tablespace y un esquema. Los objetos de un mismo esquema pueden estar dispuestos en distintos tablespaces, y un tablespace puede tener objetos de distintos esquemas.

Oracle permite la utilización eficiente de las estructuras lógicas de datos mediante distintos niveles de granularidad. Para ello, Oracle utiliza los *bloques* de datos, las *extensiones* y los *segmentos*.

Al nivel más detallado, los datos de la Base de Datos se guardan en *bloques*. Un bloque de datos se corresponde con un número de bytes del nivel físico.

El siguiente nivel de detalle para espacios se denomina *extensión*. Una extensión es un número de bloques de datos contiguos, y se utilizan para guardar un tipo de información específico.

El siguiente nivel de detalle para el almacenamiento lógico se corresponde con los *segmentos*. Un segmento es un conjunto de extensiones asignados a una estructura lógica concreta. De esta forma pueden distinguirse entre varios tipos de segmentos: de datos, de índices y de rollback.

Estructura Física de la Base de Datos

Las estructuras físicas de una base de datos Oracle son los archivos de datos, los archivos de registro de operaciones (*logs*) y los archivos de control.

Cada base de datos tiene uno o más *archivos físicos de datos*. Los archivos de datos contienen todos los datos de la Base de Datos. Los datos de las estructuras lógicas de la Base de Datos, como son las tablas y los índices, se guardan físicamente en los archivos de datos.

Cada Base de Datos tiene dos o más *archivos de registro de operaciones*. La función principal de estos archivos es la de guardar todos los cambios realizados sobre los datos. De esta forma, si ocurre un error al escribir en los archivos de datos, los cambios se pueden recuperar de los archivos de registro de operaciones. Oracle dispone de varias copias de estos archivos, de forma que si uno falla, se utilizaría otro.

Cada Base de Datos tiene un *archivo de control* con información específica sobre la estructura física de la base de datos. Suele contener el nombre de la Base de Datos, los nombres y las rutas de acceso a de los archivos de datos y de los archivos de registro de operaciones, y la fecha de creación de la Base de Datos.

A.4.3. ESTRUCTURA DE MEMORIA Y PROCESOS

Estructuras de memoria

Oracle crea y utiliza diferentes estructuras de memoria para realizar varios trabajos: el *área global del sistema* (que incluye a los buffers de la Base de Datos, los buffers de registro de operaciones, y la zona de paginación compartida) y las *áreas globales de programa*.

El *área global del Sistema (SGA)* es una región de memoria compartida que contiene información sobre datos y control de una instancia Oracle. Un SGA y los procesos de background forman una instancia Oracle.

Oracle reserva el área global del sistema cuando se inicia una instancia y se libera cuando se finaliza la instancia. Cada instancia tiene su propio SGA, y los usuarios que estén conectados a un servidor Oracle comparten los datos en el SGA.

La información que se guarda en el SGA se guarda en distintos tipos de estructuras de datos. Estas estructuras de datos se definen al crear la base de datos, y se hace con un tamaño por omisión:

- *Buffer caché de la Base de Datos:* guarda los bloques usados más recientemente. Esto reduce el número de accesos a disco aumentando el rendimiento de la Base de Datos.
- *Buffer de registro de operaciones:* guarda un registro de las modificaciones realizadas sobre la Base de Datos. Estos buffers se guardan en un archivo de registro de operaciones, que son los que se utilizan en la recuperación de Bases de Datos.
- *Zona de paginación compartida:* parte del SGA que contiene zonas de memoria compartida, como son las zonas SQL compartidas (contienen el árbol de ejecución de una consulta y permite su reutilización).

El *área global de Programa (PGA)* contiene la información de control para un proceso del servidor.

Procesos de Oracle

Oracle crea un conjunto de procesos de background para cada instancia. Un SGA y un conjunto de procesos background de Oracle forman una instancia Oracle. Los procesos de background de Oracle son los siguientes:

- *Database Writer* (DBWR Escritor de la Base de Datos). Escribe en los archivos de datos los bloques modificados a partir del buffer de la Base de Datos.
- *Log Writer* (LGWR Escritor del registro). Escribe en disco las entradas del registro de operaciones.
- *Checkpoint* (CKPT). A menudo hay que escribir (DBWR) en los archivos de datos todos los buffers modificados del SGA. El proceso Checkpoint es el encargado de llamar a DBWR indicándole que actualice los archivos de datos.
- *System Monitor* (SMON Monitor del sistema). Realiza la recuperación de instancias durante el inicio de una instancia. SMON también libera los segmentos temporales que ya no se utilizan y recupera transacciones determinadas debido a algún tipo de error del sistema.
- *Proceso Monitor* (PMON Monitor de procesos). Recupera los procesos cuando falle un proceso de usuario.
- *Archiver* (ARCH Archivador). Copia los buffers de registro en los archivos de registro.
- *Recoverer* (RECO Recuperador). Resuelve transacciones distribuidas pendientes debidos a errores de red o de sistema. Actúa de forma temporizada, y conecta las Bases de Datos remotas para confirmar o deshacer transacciones pendientes.
- *Dispatcher* (Dnnn). Son procesos background optativos y son los responsables de enrutar las peticiones a los procesos de servidor compartidos y devolver las respuestas a los procesos de usuarios correspondientes.
- *Lock* (LCKn). Bloquean Inter.-instancias en servidores paralelos.

A.5. UML (LENGUAJE DE MODELADO UNIFICADO)

UML es un lenguaje gráfico para modelar sistemas software. Fue creado a principios de los 90 por tres líderes en el mundo de modelado de objetos: Grady Booch, James Rumbaugh e Ivars Jacobson. Su propósito era unificar los tres lenguajes de modelo que existían en uno. UML es ahora un estándar del Grupo de Gestión de Objetos (OMG).

La arquitectura y diseño de un sistema nos permite tener diferentes puntos de vista del sistema y controlar el conjunto del proceso de desarrollo del sistema. UML nos permite modelar un sistema según varias perspectivas o vistas (véase la Fig. A.1.).

Los diagramas generados a partir de los procesos de gestión del proyecto serán los artefactos perdurables que documentarán el propósito, ámbito, estructura, comportamiento, implantación y demás aspectos del proyecto.

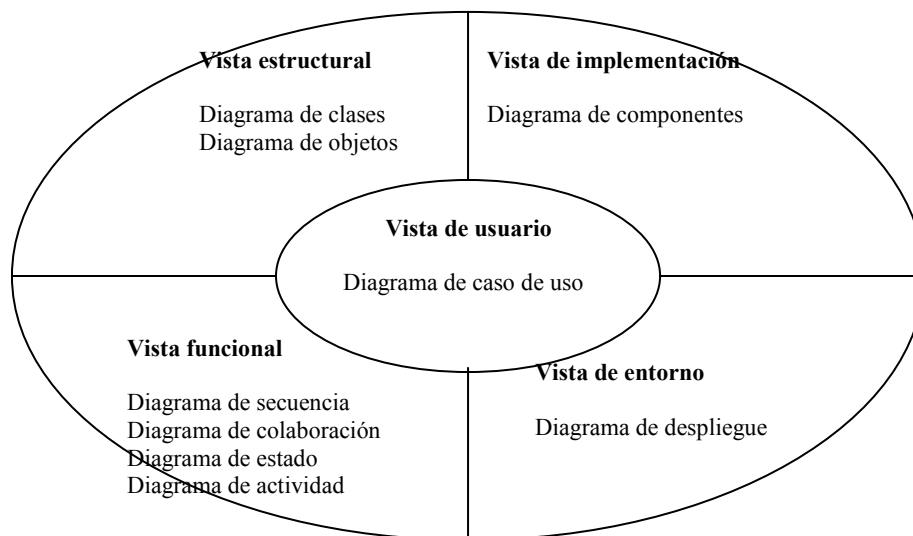


Figura A.1. Vistas y diagramas asociados en UML

Vista de usuario

La vista de usuario representa al sistema desde la perspectiva de los usuarios del sistema. Un Diagrama de Casos de Uso representa la funcionalidad provista por el

sistema de cara a los usuarios exteriores. Está compuesto de actores, casos de uso y relaciones entre ellos.

Vista estructural

La vista estructural representa el aspecto estático del sistema, y está representada fundamentalmente por los Diagramas de Clase, compuestos de clases y relaciones entre ellas.

Por otra parte, los Diagramas de Objetos representan la estructura estática de un sistema en un momento definido de tiempo. Estos diagrama están compuestos de nodos objeto y asociaciones.

Vista funcional

La vista funcional representa el aspecto dinámico del sistema. Un Diagrama de Secuencia representa una secuencia temporal de mensajes intercambiados entre varios objetos con el fin de conseguir un comportamiento específico.

Un Diagrama de Colaboración representa un comportamiento particular distribuido entre varios objetos. Estos diagramas están compuestos por objetos, sus asociaciones y los intercambios de mensajes que permiten cumplir este comportamiento.

Un Diagrama de Estados representa los estados y respuestas de una clase a sucesos externos e internos. Puedo ser utilizado también para representar el ciclo de vida de un objeto.

Un Diagrama de Actividad representa las actividades o acciones de un proceso sin importar los objetos que desempeñen dichas actividades.

Vista de implementación

La vista de implementación representa los aspectos estructurales y funcionales de la realización del sistema. Un Diagrama de Componentes representa la configuración y las dependencias entre componentes implementados por software.

Vista del entorno

La vista del entorno representa al medio físico en el cual se ha realizado el sistema. Un Diagrama de Despliegue representa la red de elementos procesadores de recursos y la configuración de los componentes de software de cada elemento físico.

A continuación se detallan de forma más exhaustiva los componentes de algunos de los diagramas más importantes de UML, para la comprensión del lector de los diagramas incluidos en el proyecto:

Los **Diagramas de Casos de Uso** representan las funcionalidades que el sistema proporciona a los usuarios externos. El Diagrama de Casos de Uso está compuesto por actores, nodos de casos de uso, y sus correspondientes relaciones. Los actores pueden ser personas que utilizan la aplicación u otros sistemas.

Un nodo de caso de uso se representa mediante un óvalo etiquetado. La etiqueta indica un resumen de la actividad que el sistema desempeña para el actor.

El **Diagrama de Clases** presenta de una manera estática la estructura de información del sistema. Estos diagramas contienen, entre otros, los siguientes elementos:

- *Clases*, que representan objetos con características y comportamientos similares. Estas características incluyen los atributos, que describen las características propias del objeto, las operaciones, que describen el comportamiento, y las asociaciones, que representan relaciones entre clases. La representación depende del nivel de detalle del diagrama, tal y como puede verse en la siguiente figura:

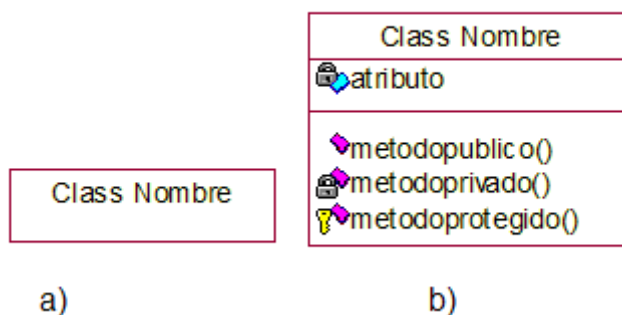


Figura A.2. Tipos de notación para una clase

La representación más simple consiste en un rectángulo con el nombre de la clase en su interior (Figura A.8). Cuando se especifican los atributos y/o métodos de la clase, el rectángulo anterior se divide en varios departamentos. El segundo compartimento se utiliza para los atributos. El tercer compartimento se utiliza para los métodos. Los métodos públicos van precedidos por el símbolo (`+`), los métodos protegidos de (`+`), y los métodos privados de (`-`).

- *Asociaciones*, que permiten modelar las relaciones existentes entre las distintas clases. Se denotan como líneas que conectan dos o más clases. Pueden tener nombre que identifique a la relación, dirección y tener roles desempeñados por las clases relacionadas. Un ejemplo de asociación puede verse en la siguiente figura:



Figura A.3. Tipos de notación para una asociación

- *Generalizaciones*, que son asociaciones entre elementos genéricos y específicos. Los elementos más específicos heredan y pueden modificar y/o añadir características a los proporcionados por los elementos más genéricos. La representación puede hacerse de la forma expuesta en la Figura A.4.

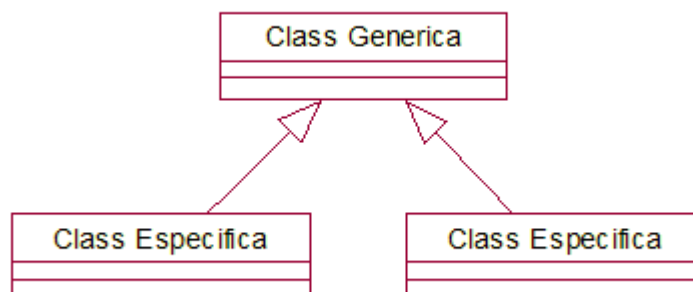


Figura A.4. Tipos de notación para una generalización

- *Agregaciones* que permiten modelar una asociación entre una clase y las clases que lo componen. La forma de representación de esta asociación puede verse en la siguiente figura:



Figura A.5. Tipos de notación para una agregación

Un **Diagrama de Componentes** representa la organización y dependencias entre componentes software. En la siguiente figura se muestra el icono correspondiente para representar los componentes software de un sistema.



Figura A.6. Nodo de componente

Un **Diagrama de Despliegue** representa la red de elementos procesadores y la configuración de componentes de software en cada elemento físico. Este diagrama se compone de nodos de hardware, software, componentes, dependencias de software y relaciones de comunicación. Este tipo de diagrama puede utilizarse para mostrar cómo están configuradas las capas lógicas de la arquitectura de una aplicación dentro de la red física.

ANEXO B

MANUAL DE USUARIO

La aplicación presenta un interfaz gráfico. A continuación se presentan y describen las ventanas de la aplicación.

B.1. Acceso a la aplicación

Para realizar cualquier modificación en el sistema, el usuario antes debe validarse para que la aplicación compruebe si tiene permisos para acceder a la aplicación y cuáles son éstos.



Figura B.1. Validación

Una vez hayamos hecho una autenticación válida accederemos al menú de gestión.



Figura B.2. Menú gestión

B.2. Gestión de espectáculos

En esta sección se verá cómo administrar la información de los espectáculos, para ello se puede navegar por los diferentes enlaces del menú superior.

Insertar un espectáculo nuevo

Al elegir esta opción se nos permitirá insertar un espectáculo nuevo, en esta primera pantalla se podrá rellenar la información general del espectáculo: título, descripción breve, información, elenco, equipo artístico, duración etc. Además se podrá seleccionar la categoría, la calificación por edades, la compañía y el género.

Una vez tengamos todos los datos introducidos pulsaremos en “enviar”.



The screenshot shows the 'Auditorios • Teatros • Espectáculos' web application. The main navigation bar includes links: 'Insertar un espectáculo nuevo', 'Modificar un espectáculo', 'Borrar espectaculo', and 'Mostrar espectaculo'. Below this is a secondary navigation bar with 'Gestión de contenidos', 'Index', 'Gestión de páginas', and 'Gestión de eventos'. The form for 'Insertar un espectáculo nuevo' contains the following fields:

- Título del espectáculo:** Text input with 'La cena de los generales'.
- Compañía:** Dropdown menu with 'Seleccione una compañía' and a button 'Añadir nueva compañía' with 'producciones faraute'.
- Categoría:** Radio buttons for 'Invierno' (selected), 'Otoño', 'Primavera', and 'Teatro Real'.
- Género:** Dropdown menu with 'Teatro' and a button 'Añadir nuevo género'.
- Calificación:** Dropdown menu with 'Todos los públicos' and a button 'Añadir nueva calificación'.
- Subtítulo:** Text input.
- Duración:** Text input with '90'.
- Descripción:** Text area with 'Afirma Narros que "tenemos la necesidad y estamos condenados a entendernos. Somos los encargados de reconstruir la convivencia, por eso la cocina en la obra representa a la España devastada que hay que poner en orden."'.
- Información:** Text area with the same text as the description.
- Elenco Artístico:** Text area with 'Emilio Gómez, Víctor Manuel Dogar, César Oliver, Luis Muñiz, Adolfo de Grandy, Ana Goya, Candela Arroyo, Juan de Mata, Lucía Bravo, Virginia Mateo, Luis Garbayo, Borja Luna y Tomás Calleja'.
- Dirección:** Text input with 'MIGUEL NARROS'.
- Equipo técnico:** Text area.
- Venta de Entradas:** Text input.
- Enlace Entradas.com:** Text input.

At the bottom of the form are two buttons: 'Limpiar' and 'Enviar'.

Figura B.3. Insertar nuevo espectáculo

Una vez introducida toda esta información se podrán elegir los datos que se quieran rellenar entre fechas, precios, imágenes o enlaces de interés. Además también se podrán modificar datos ya añadidos.



Auditorios • Teatros • Espectáculos

Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo

Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos

Indique qué desea hacer:

- Insertar fechas
- Insertar Enlaces
- Insertar Precios
- Insertar imágenes
- Insertar archivos
- Modificar datos añadidos

Figura B.4. Inserción de datos de espectáculo

En este caso elegiremos insertar fechas y podremos añadir tantas como queramos. Una vez hayamos introducido todas pulsamos en “Insertar más datos”.



Auditorios • Teatros • Espectáculos

Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo

Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos

Nombre del espectáculo: La cena de los generales

Actuación nº	Día	Mes	Año	Hora	Minutos
1	16	Enero	2010	10	0
2	12	Enero	2010	17	30
3	1	Enero	2009	0	00

Insertar Fechas

Insertar más datos

Figura B.5. Inserción de fechas

Si queremos añadir enlaces de interés en la ficha del espectáculo pulsaremos en “Insertar enlaces” y tendremos que rellenar la siguiente pantalla:



The screenshot shows a web interface for managing content. At the top, there's a header with the title "Auditorios • Teatros • Espectáculos" and a navigation bar with links: "Insertar un espectáculo nuevo", "Modificar un espectáculo", "Borrar espectáculo", and "Mostrar espectáculo". Below this is another navigation bar with "Gestión de contenidos", "Index", "Gestión de páginas", and "Gestión de eventos". The main form area is titled "Insertar enlace" and contains the following fields and controls:

- "Seleccionar enlace ya añadido": A dropdown menu with "Seleccione un enlace" and a downward arrow.
- "Añadir enlace nuevo": A section containing:
 - "Nombre": A text input field with "Producciones Faraute" entered.
 - "url": A text input field with "http://www.produccionesfaraute.cc" entered.
 - "¿Guardar en lista de enlaces?": A checkbox that is currently unchecked.
- Buttons: "Limpiar", "Insertar enlace", and "Insertar más datos".

Figura B.6. Inserción de enlaces

Para insertar los precios del espectáculo pulsaremos en Insertar precios, obtenemos una pantalla en la que se nos pregunta el número de zonas de precios. Se podrá elegir entre una, dos o espectáculo gratuito.



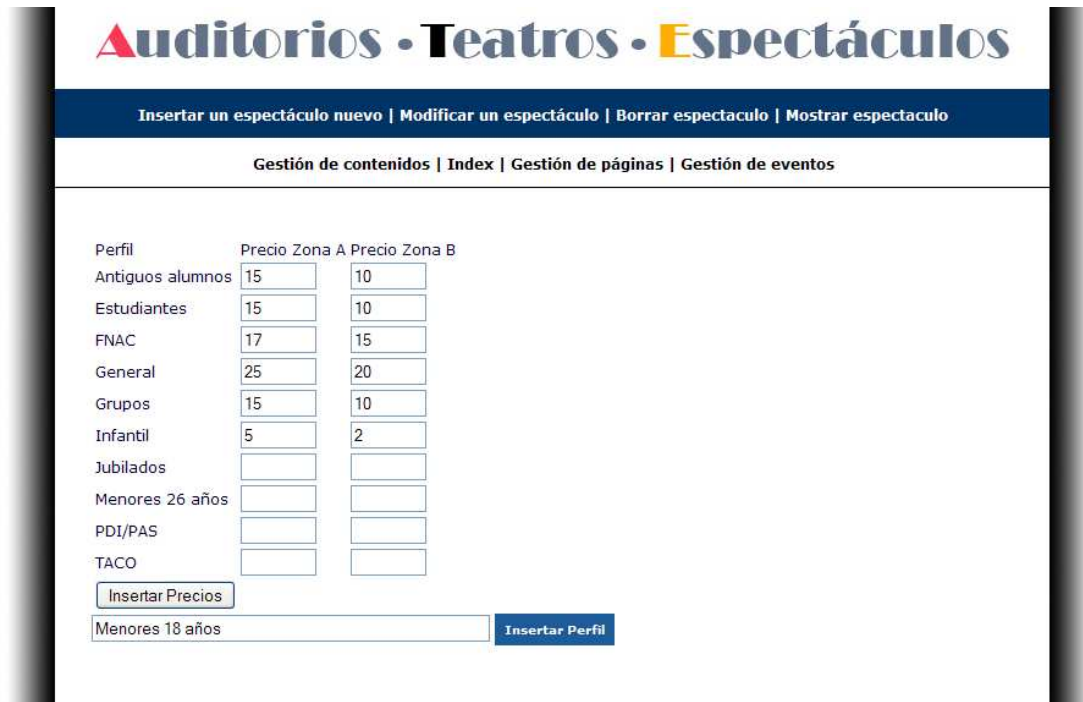
The screenshot shows a web interface for selecting price zones. It has the same header and navigation bar as Figure B.6. The main form area is titled "Seleccionar zonas de precios" and contains three radio button options:

- "Zona Única": The selected option.
- "Doble zona": An unselected option.
- "Espectáculo gratuito": An unselected option.

Figura B.7. Seleccionar zonas de precios

Si elegimos dos zonas obtendremos la siguiente pantalla en la que aparecen todos los perfiles usados hasta el momento y una zona para insertar cada precio por zona y

perfil. Si deseamos añadir algún perfil que no exista lo podremos insertar y quedará guardado para otros espectáculos.



Perfil	Precio Zona A	Precio Zona B
Antiguos alumnos	15	10
Estudiantes	15	10
FNAC	17	15
General	25	20
Grupos	15	10
Infantil	5	2
Jubilados		
Menores 26 años		
PDI/PAS		
TACO		

Menores 18 años

Figura B.8. Inserción de precios

Para insertar una imagen pulsamos en insertar imágenes, rellenamos los campos, adjuntamos la imagen y elegimos si nuestra imagen es para la ficha o para las páginas generales de programación. Una vez completados todos los campos, pulsaremos en “Insertar imagen”.



Título
 ALT
 Alto px Ancho px
 Tamaño Kb
 Imagen
 Ubicación ☒ ficha ☐ hoy ☐ indice
 Copyright

Figura B.9. Inserción de imágenes

Para insertar archivos rellenaremos el formulario hasta insertar tantos archivos como necesitemos. Cuando terminemos pulsamos en insertar campos.



The screenshot shows the 'Auditorios · Teatros · Espectáculos' web application. At the top, there is a navigation bar with links: 'Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo'. Below this is a secondary navigation bar with links: 'Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos'. The main content area contains a form for inserting a new spectacle. The form fields are: 'Titulo' (text box with 'Dossier'), 'Tamaño' (text box with '153' and 'Kb' next to it), 'Archivo' (text box with 'C:\Documents and Settings\rffraile' and a 'Browse...' button), and 'Formato' (text box). Below the form fields are three buttons: 'Limpiar', 'Insertar archivo', and 'Insertar campos'.

Figura B.10. Inserción de archivos

De esta manera ya estará insertada toda la información referente al espectáculo.

Modificar espectáculos

Una vez insertada toda la información en el espectáculo podemos elegir la opción modificar datos añadidos y podremos elegir el campo a modificar. En caso de que el cambio sea posterior podremos elegir la opción del menú superior "modificar un espectáculo". En cualquiera de los casos obtendremos la siguiente pantalla:



Auditorios • Teatros • Espectáculos

Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo

Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos

¿Qué espectáculo desea modificar?

Seleccione un espectáculo ▼

Selecciona el campo que quieres modificar

Archivos ▼ **Modificar espectáculo**

Figura B.11. Seleccionar espectáculo

Una vez hayamos elegido el espectáculo a modificar y el campo que queramos cambiar pulsamos en Modificar espectáculo.

En la pantalla resultante haremos la modificación que necesitemos y pulsamos en Hacer otra modificación. Podemos realizar esta operación en todos los campos que necesitemos.



Auditorios • Teatros • Espectáculos

Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo

Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos

Nombre del espectáculo: La cena de los generales

Contenido actual de Genero: Teatro

Nuevo contenido

Cine ▼ **Modificar Genero**

El espectáculo pertenece a un nuevo Genero

Insertar Genero

Hacer otra modificación

Figura B.12. Modificar espectáculos

Eliminar espectáculos

Para borrar un espectáculo que ya no pertenezca a la programación pulsaremos “borrar espectáculo” en el menú superior. Para proceder a la eliminación únicamente hay que seleccionar el espectáculo que queramos borrar y pulsamos en “borrar espectáculo”.



Figura B.13. Eliminar espectáculo

B.3. Gestión de contenidos

Si queremos modificar cualquiera de los campos que aparecen en los desplegables comunes a varios espectáculos como género, calificación, categoría etc. Seleccionaremos la opción “Gestión de contenidos” del menú horizontal. Una vez hagamos esto, elegiremos el campo a modificar y pulsaremos en modificar.



Figura B.14. Seleccionar tipo de contenido

En cualquiera de los campos que elijamos obtendremos una pantalla similar:



The screenshot shows the top navigation bar with the title 'Auditorios • Teatros • Espectáculos' and a menu with 'Insertar un espectáculo nuevo', 'Modificar un espectáculo', 'Borrar espectáculo', and 'Mostrar espectáculo'. Below this is a secondary menu with 'Gestión de contenidos', 'Index', 'Gestión de páginas', and 'Gestión de eventos'. The main content area displays 'Campo seleccionado: Calificación'. There is a dropdown menu currently showing 'Entre 6 y 11 años'. To the right of the dropdown are three radio buttons labeled 'Modificar', 'Borrar', and 'Agregar'. Below these elements is a blue button labeled 'Modificar/borrar campo'.

Figura B.15. Seleccionar contenido

En esta pantalla seleccionaremos el valor a modificar o borrar y pulsaremos en el botón modificar/borrar campo. En el caso de que queramos añadir uno nuevo seleccionaremos agregar.

B.4. Gestión de Eventos

En caso de que queramos agregar, borrar o modificar un evento pulsaremos en la opción del menú superior “gestión de eventos”.



This screenshot shows the same top navigation and secondary menu as Figure B.15. The main content area is titled 'Gestión de eventos'. It features a dropdown menu with the placeholder text 'Seleccione el evento'. To the right of the dropdown are two radio buttons labeled 'Modificar' and 'Borrar'. Below these are two blue buttons: 'Modificar/borrar campo' and 'Agregar evento nuevo'.

Figura B.16. Gestión de eventos

Insertar eventos

Para insertar un evento pulsaremos en el botón “Agregar evento nuevo”.



The screenshot shows the 'Auditorios • Teatros • Espectáculos' web application. The header includes the title and navigation links: 'Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo'. Below this is a secondary navigation bar with 'Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos'. The main form for inserting a new event contains the following fields:

- Nombre:** Lanzamiento de la nueva aplicación
- Enlace:** http://www.ate.es
- Imagen:** D:\Mis documentos\Pro (with a 'Browse...' button)
- Fecha de publicación:** 15 / Diciembre / 2009 10 : 10
- Fecha de baja:** 16 / Febrero / 2010 10 : 10

At the bottom of the form is a blue button labeled 'Insertar Evento'.

Figura B.17. Insertar evento

Una vez hayamos rellenado todos los campos pulsaremos en “Insertar evento” y el evento quedará añadido.

Modificar eventos

Para modificar un evento, seleccionamos el evento que queremos modificar, y obtendremos una pantalla similar a la de inserción con los campos rellenos con su información. Realizaremos las modificaciones y pulsamos en modificar.



The screenshot shows the 'Auditorios • Teatros • Espectáculos' web application. At the top, there is a navigation bar with links: 'Insertar un espectáculo nuevo', 'Modificar un espectáculo', 'Borrar espectáculo', and 'Mostrar espectáculo'. Below this is another navigation bar with links: 'Gestión de contenidos', 'Index', 'Gestión de páginas', and 'Gestión de eventos'. The main content area displays a form for modifying an event. The form fields are: 'Nombre' (Lanzamiento de la nueva aplicación), 'Enlace' (http://www.ate.es), 'Imagen' (D:\Mis documentos\Pro with a 'Browse...' button), 'Fecha de publicacion' (15 / Diciembre / 2009 10 : 10), and 'Fecha de baja' (16 / Febrero / 2010 10 : 10). A 'Modificar Evento' button is located at the bottom left of the form.

Figura B.18. Modificar eventos

Borrar eventos

Para borrar un evento, únicamente tendremos que seleccionar el evento a eliminar, marcar borrar y a continuación pulsar en “borrar/modificar evento”.

B.5. Gestión de páginas

Para poder realizar acciones sobre las páginas de contenido general, primero hay que pulsar en “Gestión de páginas” en el menú superior. Aparecerá la siguiente página:



Auditorios · Teatros · Espectáculos

Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo

Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos

Por favor, indique la acción a realizar

☐ Modificar ☐ Borrar ☐ Agregar pagina

Seleccione la página (en caso de modificar o suprimir)

- ☐ Menu primer orden
- ☐ Menu segundo orden
 - ☐ cómo llegar
 - ☐ En el coche de san fernando
 - ☐ En coche
 - ☐ M-40
 - ☐ En transporte público
 - ☐ A pedales
 - ☐ públicos y servicios
 - ☐ Venta de localidades

Aceptar

Figura B.19. Gestión de páginas

En ella podremos elegir si queremos modificar, borrar o crear una página nueva. Debemos seleccionar la opción que necesitemos.

Insertar página

Si queremos añadir una página nueva, seleccionaremos “Agregar página”.



The screenshot shows a web interface for managing content. At the top, there's a header with the logo 'Auditorios • Teatros • Espectáculos'. Below it, a navigation bar contains links: 'Insertar un espectáculo nuevo | Modificar un espectáculo | Borrar espectáculo | Mostrar espectáculo'. Underneath, another navigation bar lists 'Gestión de contenidos | Index | Gestión de páginas | Gestión de eventos'. The main form area has three fields: 'Nombre' with the value 'Prensa', 'Imagen' with a 'Browse...' button, and 'Contenido' with a text area containing the text 'En esta página se colgarán notas de prensa e información relativa a actividades para su difusión.' At the bottom left of the form is a blue button labeled 'Insertar Página'.

Figura B.20. Insertar página

Una vez rellena la información pulsamos en “Insertar página”. A continuación aparecerá un menú en el que habrá que elegir la página padre de nuestra nueva página. En caso de que queramos que aparezca en el menú horizontal debemos indicar en cuál de los dos queremos que aparezca.



The screenshot shows the same web interface as Figure B.20, but after clicking 'Insertar Página'. The 'Contenido' text area is now empty. Below the text area, a section titled 'Seleccione la página padre' displays a list of menu items with expandable/collapsible icons (plus and minus signs). The items are: 'Menu primer orden', 'Menu segundo orden', 'cómo llegar' (with sub-items 'En tren', 'En coche', and 'M-40'), 'En Metro', 'públicos y servicios', and 'Venta de localidades'.

Figura B.21. Seleccionar página padre

Una vez seleccionada la ubicación se podrá seleccionar la posición de la página dentro del menú. Para ello en la siguiente ventana deberemos elegir la posición y pulsar el botón “Seleccionar posición”.

Modificar página

En caso de que queramos modificar una página seleccionaremos “modificar” y a continuación la página en la que queramos hacer algún cambio. A continuación pulsaremos en “Aceptar”.



The screenshot shows a web application interface for managing content. At the top, there is a header with the title "Auditorios • Teatros • Espectáculos" in a stylized font. Below the header, there is a navigation bar with links: "Insertar un espectáculo nuevo", "Modificar un espectáculo", "Borrar espectáculo", and "Mostrar espectáculo". Below this, there is a sub-navigation bar with links: "Gestión de contenidos", "Index", "Gestión de páginas", and "Gestión de eventos". The main content area contains a form for modifying a page. The form has three fields: "Nombre" with the value "En tren", "Imagen" with a "Browse..." button, and "Contenido" with a text area containing the text "Línea 5 de cercanías". At the bottom of the form, there is a blue button labeled "Modificar Página".

Figura B.22. Modificar página

Una vez hayamos realizado todas las modificaciones necesarias pulsamos en “Modificar página”.

Borrar página

Para borrar seleccionamos la opción “borrar”, la página que queramos borrar y a continuación pulsaremos en “Aceptar”.

B.6. Gestión de usuarios

El usuario administrador tiene permisos para agregar, modificar o eliminar otros gestores, ya sean administradores o usuarios avanzados. Para hacer esto, este usuario tiene en su menú la opción Gestión de usuarios. Una vez pulsemos en esta opción veremos la siguiente pantalla.



Figura B.23. Gestión de usuarios

Para crear un nuevo usuario pulsaremos en Alta usuario. Rellenamos todos los campos y pulsamos a “Enviar”.



Figura B.24. Insertar usuario

Para modificar los datos de un usuario pulsamos en “Modificar usuario” y debemos seleccionar el usuario en la siguiente pantalla.



Figura B.25. Seleccionar usuario

Una vez pulsemos en “Modificar” modificaremos los campos que queramos y pulsamos en “Enviar”.



Figura B.26. Modificar usuario

A la hora de borrar un usuario el procedimiento es similar: buscaremos el usuario que queramos eliminar y pulsaremos en borrar.

B.7. Cerrar sesión

Una vez hayamos hecho todas las inserciones/modificaciones oportunas pulsaremos en cerrar sesión.

B.8. Búsquedas

La página programación permite hacer búsquedas de espectáculos por géneros, calificación por edades, compañía, cursos, y categoría. Podemos realizar tantas búsquedas como queramos con la ayuda de los desplegaables permitiendo hacerlas además por varios campos a la vez.



Auditorios • Teatros • Espectáculos

Prensa | Alquiler de espacios | Gestión

programación | Venta de localidades | públicos y servicios | cómo llegar

Temporada ▼ Curso ▼ Ballet de Mo: ▼ Danza ▼ Calificación por ▼ Buscar

La cenicienta
Ballet de Moscú
Danza
Una vez más el Teatro Gran Vía abrirá sus puertas a los amantes de la Danza, ofreciendo los repertorios más distinguidos del Ballet Clásico De Moscú de N. Kasatkina y V. Visilyov, tras la excelente acogida del público Madrileño el pasado año. En esta ocasión serán siete únicas semanas en la que la compañía representará cuatro de los repertorios más aplaudidos de la danza clásica: La Bella Durmiente, La Traviata, La Cenicienta y Giselle.

Figura B.27. Realización de búsquedas